



**GUILHERME FERREIRA TORRES**

**DESENVOLVIMENTO DE UM MODELO DE  
BANCO DE DADOS PARA UM AMBIENTE  
*ONLINE* DE GERÊNCIA DE PROJETOS**

**LAVRAS - MG  
2012**

**GUILHERME FERREIRA TORRES**

**DESENVOLVIMENTO DE UM MODELO DE BANCO DE DADOS  
PARA UM AMBIENTE *ONLINE* DE GERÊNCIA DE PROJETOS**

Monografia apresentada ao colegiado do Curso de Sistemas de Informação, como uma das exigências para a obtenção do título de Bacharel em Sistemas de Informação.

Orientadora:

Dr. Ana Paula Piovesan Melchiori

Co-orientadora:

M.Sc. Rosana Áurea Tonetti Massahud

**LAVRAS - MG  
2012**

**GUILHERME FERREIRA TORRES**

**DESENVOLVIMENTO DE UM MODELO DE BANCO DE  
DADOS PARA UM AMBIENTE ONLINE DE GERÊNCIA  
DE PROJETOS**

Monografia de graduação apresentada ao  
Colegiado do Curso de Sistemas de  
Informação, para obtenção do título de  
Bacharel em Sistemas de Informação.

APROVADA em 10 de outubro de 2012.

ANDRÉ GRUTZMANN

ANDRÉ LUIZ ZAMBALDE

  
ANA PAULA PIOVESAN MELCHIORI (orientador/a)

ROSANA ÁUREA TONETTI MASSAHUD (Co-Orientador)

**LAVRAS-MG**

**2012**

*Aos meus pais, Mauro e Cleide, a minha irmã Maíra, a minha companheira*

*Danielle, a meus avós João, Hemerita, José e Anália.*

*Aos meus amigos e minha toda minha família.*

**Dedico.**



## **AGRADECIMENTOS**

Agradeço a Deus pela saúde, proteção e pelas suas bênçãos. Aos meus pais Mauro e Cleide, por todo o apoio e incentivo durante essa caminhada. A minha irmã Maíra, pelo companheirismo. A minha namorada Danielle, por seu incansável carinho e compreensão. Aos meus queridos avós João, Emerita, José e Anália, pelos seus inúmeros exemplos de carisma e amor. A todos os meus primos e amigos, a quem considero verdadeiros irmãos, pelos incontáveis momentos de união e companheirismo. Aos meus companheiros de sala, meus eternos amigos que ficarão para sempre no meu coração. A todas as pessoas com quem trabalhei na Devex, pela confiança, profissionalismo e amizade. A todos que acreditaram e confiaram na realização desse sonho, o meu muito obrigado.

Agradeço também aos professores que contribuíram para o enriquecimento de meu conhecimento. Obrigado em especial a minha orientadora Ana Paula e a minha co-orientadora Rosana, por toda a dedicação no auxílio desse trabalho. A todos da banca, obrigado pela disposição e apoio.

Um grande abraço a todos!

## RESUMO

Com o surgimento de novas tecnologias e a crescente dependência do mercado a elas, a indústria de tecnologia da informação vem obtendo um enorme crescimento em sua demanda de serviços. Inúmeros projetos de grande relevância e complexidade são cada vez mais constantes e desafiadores. Por esse motivo, a gerência de projetos se torna tão importante para o sucesso dos projetos. Atualmente, a utilização de ambientes de gerência de projetos tem se tornado essencial para o sucesso dos projetos, pois auxilia o usuário em todo o ciclo de vida do projeto. O objetivo deste trabalho foi criar um modelo de banco de dados que sirva como base para criação de uma nova ferramenta de gerência de projetos. Para tal objetivo, o trabalho implicou em uma pesquisa qualitativa, a qual foi composta de uma análise comparativa dos ambientes DotProject, NetOffice e ClockingIt, com o objetivo de buscar conhecimento nesse contexto para utilizá-lo na criação do novo modelo de banco de dados. O resultado obtido foi a criação de um modelo de banco de dados no MySQL e a elaboração de um dicionário de dados com as características de cada objeto. Desenvolver uma nova ferramenta em cima do modelo criado pode ser vantajoso pela possibilidade de modelar o banco de dados de acordo com as necessidades da nova ferramenta de gerência de projetos.

Palavras-chave: Ambientes de Gerencia de Projetos. Modelagem de Dados. Banco de Dados.

## **ABSTRACT**

With the emergence of new technologies and the ascending dependence of the business to them, the information technology industry has achieved an enormous growth in demand for their services. Many projects of great importance and complexity are increasingly constant and challenging. For this reason, the project management is so important to the success of projects. Currently, the use of project management software has become essential for the success of projects, because it assists the user in the lifecycle of the project. The objective was to create a model database which serves as basis for creating a new project management tool. For this objective, the work resulted in a qualitative research, which consisted of a comparative analysis of software's DotProject, NetOffice and ClockingIt, with the goal of seeking knowledge in this context to use it in the creation of the new database model. The obtained result was the creation of a database model in MySQL and the elaboration of a data dictionary with the characteristics of each object. Develop a new tool upon the model created can be advantageous by the possibility of modeling the database according the needs of the new project management tool.

**Keywords:** Project Management Tool. Data Modeling. Database.

## LISTA DE FIGURAS

Figura 1- Fases de um projeto de <i>Software</i> , baseado no PMBOK.....	18
Figura 2 - Representação de uma tabela no modelo dados relacional.....	26
Figura 3 - Tela de login dotProject .....	31
Figura 4 - Tela de cadastro de usuário dotProject.....	31
Figura 5 - Tela de edição de role dotProject.....	32
Figura 6 - Tela de cadastro de companhia do dotProject.....	32
Figura 7 - Tela de cadastro de projeto do dotProject .....	33
Figura 8 - Tela de cadastro de atividades do dotProject.....	34
Figura 9 - Tela de cadastro de arquivos do dotProject .....	35
Figura 10 - Parte do calendário de alocações do dotProject .....	35
Figura 11 - Criação de fórum no dotProject .....	36
Figura 12 - Criação de chamada de serviço do cliente no dotProject .....	37
Figura 13 - Tela inicial de login do ClockingIt.....	38
Figura 14 - Tela de criação de usuário do ClockingIt .....	39
Figura 15 - Tela de adição de cliente do ClockingIt .....	39
Figura 16 - Tela de adição de projeto do ClockingIt.....	40
Figura 17 - Tela de adição de meta de projeto do ClockingIt.....	40
Figura 18 - Tela de edição de permissão de projeto do ClockingIt.....	41
Figura 19 - Tela de criação de atividades de projeto do ClockingIt.....	41
Figura 20 - Tela de envio de arquivos do ClockingIt.....	42
Figura 21 - Tela calendário do ClockingIt.....	43
Figura 22 - Tela de criação de fóruns do ClockingIt.....	44
Figura 23 - Tela principal da wiki integrada do ClockingIt .....	44
Figura 24 - Tela de criação de relatório do ClockingIt .....	45
Figura 25 - Tela de criação de bate-papo do ClockingIt .....	46
Figura 26 - Tela de login do NetOffice .....	47

Figura 27 - Tela de cadastro de usuário do NetOffice.....	47
Figura 28 - Tela de cadastro de clientes do NetOffice .....	48
Figura 29 - Tela de cadastro de projetos do NetOffice .....	49
Figura 30 - Tela de cadastro de atividades do NetOffice .....	50
Figura 31 - Calendário integrado do NetOffice .....	51
Figura 32 - Criação de relatórios do NetOffice .....	51
Figura 33 - Funcionalidades disponíveis para o usuário comum.....	54
Figura 34 - Funcionalidades disponíveis para o usuário administrador .....	55
Figura 35 - Exemplo de estrutura de uma tabela no modelo UML.....	56
Figura 36 - Tipos de relacionamentos entre tabelas .....	57
Figura 37 - Relações entre tabelas de usuário.....	58
Figura 38 - Tabelas para Usuários / Permissões .....	59
Figura 39 - Relações entre tabelas de empresa .....	60
Figura 40 - Tabelas para Cadastro de Empresas .....	61
Figura 41 - Relações entre as tabelas de Cadastro de Projetos .....	62
Figura 42 - Tabelas para Cadastro de Projetos .....	63
Figura 43 - Relações entre as tabelas de Projetos .....	65
Figura 44 - Outras tabelas da funcionalidade de Projetos .....	66
Figura 45 - Relações entre tabelas da funcionalidade de tarefas .....	69
Figura 46 - Tabelas da funcionalidade de Cadastro de Tarefas .....	70
Figura 47 - Relações entre tabelas da funcionalidade de lançamento de horas ..	72
Figura 48 - Tabelas da funcionalidade de lançamento de horas .....	73
Figura 49 - Funcionamento da funcionalidade de eventos .....	75
Figura 50 - Tabelas da funcionalidade de lançamento de horas .....	76
Figura 51 - Relações da funcionalidade de Contatos .....	78
Figura 52 - Tabelas da funcionalidade de Contatos .....	78
Figura 53 - Relação da funcionalidade de Arquivos .....	79
Figura 54 - Tabelas da funcionalidade de Arquivos .....	80

Figura 55 - Representação da funcionalidade Fórum de Discussão.....	81
Figura 56 - Tabelas da funcionalidade de Fórum de Discussão.....	82
Figura 57 - Relações da funcionalidade de suporte ao cliente.....	83
Figura 58 - Tabelas da funcionalidade de Suporte ao Cliente .....	84
Figura 59 - Tabela da funcionalidade de Configurações .....	86

## LISTA DE TABELAS

Tabela 1 - Classes de objetos de um modelo entidade-relacionamento .....	22
Tabela 2 - Funcionalidades de cada ambiente .....	52
Tabela 3 - Registros default de tipos de empresas .....	61
Tabela 4 - Registros default de categorias de projeto .....	64
Tabela 5 - Registros default de status de projeto .....	65
Tabela 6 - Registros default de categoria de custo de projeto .....	67
Tabela 7 - Registros default de fases de projeto .....	71
Tabela 8 - Registros default de locais de trabalho .....	74
Tabela 9 - Registros default de tipos de eventos.....	77
Tabela 10 - Registros default de status de ocorrências.....	86
Tabela 11 - Exemplo de parâmetro da tabela de configuração .....	88

## SUMÁRIO

<b>1</b>	<b>Introdução.....</b>	<b>13</b>
<b>1.1</b>	<b>Considerações Iniciais.....</b>	<b>13</b>
<b>1.2</b>	<b>Motivação.....</b>	<b>14</b>
<b>1.3</b>	<b>Objetivos .....</b>	<b>14</b>
<b>1.4</b>	<b>Estrutura do Trabalho.....</b>	<b>15</b>
<b>2</b>	<b>Referencial Teórico.....</b>	<b>16</b>
<b>2.1</b>	<b>Projetos de <i>Software</i>.....</b>	<b>16</b>
<b>2.2</b>	<b>Gerência de Projetos de <i>Software</i> .....</b>	<b>17</b>
<b>2.3</b>	<b>Ambientes <i>Online</i> de Gerência de Projetos de <i>Software</i> .....</b>	<b>20</b>
<b>2.3.1</b>	<b>DotProject .....</b>	<b>20</b>
<b>2.3.2</b>	<b>ClockingIt .....</b>	<b>20</b>
<b>2.3.3</b>	<b>NetOffice .....</b>	<b>21</b>
<b>2.4</b>	<b>Modelagem de Dados.....</b>	<b>21</b>
<b>2.4.1</b>	<b>Modelo Conceitual.....</b>	<b>21</b>
<b>2.4.1.1</b>	<b>Modelo Entidade-Relacionamento .....</b>	<b>22</b>
<b>2.4.1.2</b>	<b>UML.....</b>	<b>24</b>
<b>2.4.2</b>	<b>Modelo Relacional .....</b>	<b>25</b>
<b>2.4.3</b>	<b>Modelo Físico .....</b>	<b>25</b>
<b>2.5</b>	<b>Banco de Dados Relacional .....</b>	<b>25</b>
<b>3</b>	<b>Metodologia.....</b>	<b>28</b>
<b>3.1</b>	<b>Tipo da Pesquisa .....</b>	<b>28</b>
<b>3.2</b>	<b>Procedimentos Metodológicos .....</b>	<b>28</b>
<b>3.3</b>	<b>Ferramentas Utilizadas.....</b>	<b>29</b>
<b>4</b>	<b>Análise dos Ambientes DotProject, ClockingIt e NetOffice.....</b>	<b>30</b>
<b>4.1</b>	<b>Análise do DotProject .....</b>	<b>30</b>
<b>4.2</b>	<b>Análise do ClockingIt .....</b>	<b>38</b>



<b>4.3</b>	<b>Análise do NetOffice .....</b>	<b>46</b>
<b>4.4</b>	<b>Comparação dos Ambientes .....</b>	<b>52</b>
<b>5</b>	<b>Desenvolvimento do Modelo do Banco de Dados para o novo ambiente de Gerência de Projetos .....</b>	<b>54</b>
<b>5.1</b>	<b>Modelagem do Banco de Dados.....</b>	<b>55</b>
<b>5.1.1</b>	<b>Multiusuário / Permissões .....</b>	<b>58</b>
<b>5.1.2</b>	<b>Cadastro de Empresas .....</b>	<b>60</b>
<b>5.1.3</b>	<b>Projetos .....</b>	<b>62</b>
<b>5.1.4</b>	<b>Tarefas .....</b>	<b>69</b>
<b>5.1.5</b>	<b>Lançamento de Horas.....</b>	<b>72</b>
<b>5.1.6</b>	<b>Eventos / Calendário.....</b>	<b>75</b>
<b>5.1.7</b>	<b>Contatos .....</b>	<b>78</b>
<b>5.1.8</b>	<b>Arquivos.....</b>	<b>79</b>
<b>5.1.9</b>	<b>Fórum de Discussão .....</b>	<b>80</b>
<b>5.1.10</b>	<b>Suporte ao Cliente .....</b>	<b>83</b>
<b>5.1.11</b>	<b>Configurações .....</b>	<b>87</b>
<b>6</b>	<b>Resultados .....</b>	<b>89</b>
<b>7</b>	<b>Conclusões e Trabalhos Futuros .....</b>	<b>90</b>
<b>8</b>	<b>Referências Bibliográficas .....</b>	<b>91</b>
	<b>Apendice A .....</b>	<b>93</b>
	<b>Apendice B .....</b>	<b>105</b>

# 1 INTRODUÇÃO

## 1.1 Considerações Iniciais

Com o surgimento de novas tecnologias e a crescente dependência do mercado a elas, a indústria de tecnologia da informação vem obtendo um enorme crescimento em sua demanda de serviços. Inúmeros projetos de grande relevância e complexidade são cada vez mais constantes e desafiadores. Por esse motivo, a gerência de projetos vem se tornando elemento essencial para o sucesso dos projetos.

A gerência de projetos pode ser considerada como uma atividade relativamente antiga, mas, apenas no início do século XX, com a criação do Gráfico de Gantt, ela passou a tomar forma. Naquela época, tratada como gerência de processos, foi aplicada ao planejamento de obras e o grande avanço obtido foi o surgimento de um cenário esperado em cada etapa do projeto durante o seu desenvolvimento.

A partir daí, vários outros métodos de planejamento surgiram, principalmente durante a II Guerra Mundial, com a criação do PERT (*Program Evolution and Review Technique*) para projetos militares e também com a criação da CPM (*Critical Path Method*) para projetos industriais (CODAS, 1987).

O gerenciamento de projetos é fator fundamental para o sucesso de um projeto. As empresas de software têm investido cada vez mais em seus processos de gerência de projetos para obterem melhor qualidade na entrega de seus produtos ou serviços. O ganho adquirido pela gerência de projetos não beneficia apenas o gerente de projetos, mas também os desenvolvedores de software, que tem mais clara suas atribuições e prazos de entrega.

Atualmente, a possibilidade de armazenamento dos dados referentes aos projetos aliado a uma boa interface possibilita uma excelente condição para

monitoramento e controle de processos. Existem hoje diversos ambientes utilizados para gerenciar projetos, alguns pagos, outros não. Muitos deles já estão em um nível de maturidade bastante avançados, mas ainda assim, nota-se a necessidade de novas funcionalidades ou até mesmo a simplificação de algumas já existentes.

## 1.2 Motivação

A motivação desse trabalho está baseada no objetivo de criação de um novo ambiente *online* de gerência de projetos de *software* capaz de oferecer objetividade, usabilidade e eficiência ao usuário.

Ter a oportunidade de desenvolver um modelo de banco de dados para um ambiente complexo de gerência de projetos é bastante desafiador e inspirador, uma vez que todos os processos de modelagem de dados precisam ser levados em consideração.

Existem diversos ambientes de gerência de projetos no mercado, mas nem todos atendem de forma efetiva os objetivos do usuário. Além disso, a maioria dos ambientes tem uma interface de difícil manuseio, tornando assim, a usabilidade do usuário bastante burocrática.

## 1.3 Objetivos

Diante do exposto, este trabalho teve como objetivo criar um modelo de banco de dados para utilização em um novo ambiente *online* de gerência de projetos.

Para construção do modelo de banco de dados foi necessário uma série de etapas que complementaram o seu processo de desenvolvimento. Essas etapas também foram consideradas como objetivos do trabalho e dentre elas estão as listadas abaixo:

- Realização uma análise comparativa entre os ambientes já existentes no mercado e com grande índice de utilização, a fim de se buscar conhecimentos sobre o contexto do trabalho.
- Construção do modelo relacional do banco de dados.

Logicamente, cada etapa percorrida durante o desenvolvimento do projeto gerou uma documentação condizente, essa mesma que foi de suma importância para o desenvolvimento do modelo de banco de dados proposto para o ambiente *online* de gerência de projetos.

#### **1.4 Estrutura do Trabalho**

O trabalho daqui em diante está estruturado em alguns capítulos. Segue uma breve explicação sobre cada um deles.

O Capítulo 1 traz uma introdução sobre o trabalho proposto.

O Capítulo 2 apresenta o Referencial Teórico, mostrando os principais conceitos utilizados no desenvolvimento do trabalho.

O Capítulo 3 mostra a metodologia adotada para o desenvolvimento do trabalho, onde será detalhado a sequência de atividades, a maneira de condução do trabalho e o modo de como os conceitos serão aplicados.

O Capítulo 4 apresenta uma análise de três ambientes de gerência de projetos, são eles: DotProject, ClockingIt e NetOffice.

O Capítulo 5 apresenta o desenvolvimento do modelo de banco de dados proposto.

O Capítulo 6 apresenta os resultados obtidos com o desenvolvimento do projeto.

Finalmente, o Capítulo 7 apresenta as conclusões extraídas a partir dos resultados obtidos e os possíveis trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

### 2.1 Projetos de *Software*

Segundo o *Project Management Body of Knowledge* (PMBOK), um projeto pode ser definido como um tipo de esforço para gerar um produto, serviço ou resultado em um determinado período de tempo.

De maneira geral, o conceito de projeto também pode ser demonstrado como uma sequência de atividades específicas que visa atingir um determinado objetivo. Um projeto depende de pessoas, processos, estruturas físicas, gerenciamento, entre outros fatores. Devido a esse grande número de variáveis, os projetos estão cada vez mais dependentes de fatores tecnológicos e gerenciais.

Segundo Ochner (2006), os projetos de *software* têm históricos marcados por prazos muito além do planejado e não cumpridos, por orçamentos estourados e, além de tudo, expectativas gerais dos clientes não satisfeitas. Todas essas dificuldades evidenciam que tende a ser muito complicado satisfazer ao mesmo tempo custo, prazo, escopo e qualidade.

Esse fato pode ser contextualizado com dados publicados por uma empresa de análise e pesquisa no setor de TI, a *Standish Group*. Segundo ela, a gerência de projetos de TI tem alguns dados bastantes preocupantes. Phillips (2003) enumerou alguns desses dados, tais como:

- 31% dos projetos chegam a ser cancelados antes de sua estimativa de conclusão;
- 88% deles ultrapassam o prazo previsto ou extrapolam o orçamento, ou ainda assim acontecem as duas situações;

- Para cada 100 inícios de projetos, há em torno de 94 reinícios deles;
- A média de estouro de orçamento chega a ser de 189%;
- A média de estouro de prazo é maior ainda, em torno de 222%.

Estes dados deixam clara a importância de uma gerência de projeto bem sucedida em um projeto de *software*. Segundo Frame (1995), o gerenciamento de projetos não está associado apenas a tecnologias, mas também a princípios da administração geral e, por esse motivo, envolve alguns fatores, tais como solução de problemas, comunicação, liderança e cultura organizacional.

Uma vez que o projeto depende da combinação de vários fatores, muitas ferramentas são utilizadas para tentativas de uma melhor gerência de projeto e, conseqüentemente, um melhor resultado obtido durante o seu desenvolvimento. Ambiente de gerência de projetos é uma ferramenta que vêm sendo bastante utilizada, pois quando bem utilizada, permite ao gerente de projetos um maior controle do andamento do projeto em tempo real, tornando-se também uma importante ferramenta de tomada de decisão.

A próxima seção tratará especificamente dos assuntos relacionados à gerência de projetos de software.

## **2.2 Gerência de Projetos de *Software***

De modo geral, a gerência de projetos pode ser definida como a capacidade de administrar e controlar tarefas cronológicas a fim de se conquistar um determinado objetivo. A gerência de projetos de TI deve ser acompanhada da necessidade de equilibrar dedicação e implementação de tecnologias conjuntamente com o papel de liderança e motivação de sua equipe (PHILLIPS, 2003).

O gerenciamento de projetos de TI é uma das atividades mais complexas, que exige uma série de cuidados a serem observados para que sejam bem sucedidos. (SCHERRER, 2009, p. 1).

O *Project Management Body of Knowledge* (PMBOK), define gerenciamento de projetos como sendo a aplicação de uma gama de conhecimentos, habilidades, ferramentas e técnicas às atividades do projeto em questão, objetivando atender as suas metas propostas.

Segundo o PMBOK (2008), o processo de gerência de projetos é dividido em cinco fases:

1. Iniciação;
2. Planejamento;
3. Execução;
4. Monitoramento e Controle;
5. Encerramento.

A relação entre essas fases é mostrada na Figura 1.



Figura 1 Fases de um projeto de *software*. Fonte: PMBOK (2008)

Conforme mostra a Figura 1, essas fases estão altamente relacionadas, sendo algumas delas totalmente dependente de outras. Por esse motivo, a gerência de projeto pode se tornar tão complexa. Algumas alterações podem refletir em determinada fase e pode acarretar mudanças também em outra

anterior ou posterior, sendo que em alguns casos essas mudanças atingem todas as fases do projeto.

Segundo Phillips (2003), antes de iniciar um projeto, deve-se ter a definição de quais objetivos serão buscados até a sua conclusão, pois um projeto só pode ser iniciado quando se sabe o que exatamente ele produzirá.

Além de uma definição clara do objetivo do projeto, faz-se necessário ter também um bom planejamento dele. Estipular tarefas, prazos, recursos, custos e outros são passos indispensáveis. Nesse contexto, os ambientes *online* de projetos são bastante eficientes, uma vez que permitem ao gerente de projetos o cadastro de tarefas com seus respectivos prazos, alocação de recursos às atividades do projeto, monitoramento e controle do desenvolvimento, suporte a sua equipe, entre outros.

Existe atualmente um grande leque de ambientes *online* de gerência de projetos, sendo que eles possibilitam aos gerentes de projetos maiores condições para cumprir com os resultados esperados pelo cliente.

Dentre os ambientes disponíveis no mercado podem-se citar três notoriamente conhecidos: DotProject, NetOffice e ClockingIt. Os dois primeiros são gratuitos e de código aberto<sup>1</sup>, já o último é apenas gratuito. Nas subseções seguintes, têm-se alguns detalhes e funcionalidades desses três ambientes de gerência de projetos.

---

<sup>1</sup> Segundo a *Open Source Initiative* (OSI, 2011), projetos de código aberto são aqueles que devem atender a 10 princípios, tais como: livre redistribuição, permissão de trabalhos derivados, não discriminação, distribuição da licença e outros.



## **2.3 Ambientes *Online* de Gerência de Projetos**

### **2.3.1 DotProject**

O dotProject é uma aplicação web que foi desenvolvida na linguagem PHP. A estrutura de banco de dados relacional utilizada é o MySQL. É uma ferramenta multiusuário, de auxílio à gestão de projetos e possui várias funções de controle de um projeto (DOTPROJECT, 2011). Além disso, a aplicação visa proporcionar ao gerente de projetos condições para gerenciar tarefas e horários, realizar a comunicação com sua equipe, compartilhar informações necessárias, entre outros. O dotProject pode ser utilizado para vários tipos de necessidade, desde as mais simples até as mais complexas.

O dotProject é usado para gerenciar o cotidiano do projeto e é capaz de definir cargas de trabalho a recursos disponíveis e que estão dispostos a cooperar para a entrega e o resultado final do projeto (DOTPROJECT, 2011).

### **2.3.2 ClockingIt**

O ClockingIt é uma solução de gerenciamento de projetos gratuita. Foi criado por Erlend e Ellen Simonsem, com contribuição de uma série de pessoas. A ferramenta foi desenvolvida na linguagem Ruby On Rails com um banco de dados em MySQL (CLOCKINGIT, 2011).

A principal motivação para criação da ferramenta se deu da necessidade de Erlend manter o controle de prazos e andamento do projeto. Atualmente, a ferramenta possui diversas funcionalidades que possibilitam ao gerente do projeto conseguir acompanhar seu andamento em tempo real.

### **2.3.3 NetOffice**

O NetOffice é outra aplicação *online*, gratuita, desenvolvida em PHP e com estrutura de banco de dados relacional em MySQL. A ferramenta tem como objetivo de permitir o controle e monitoramento do projeto ao longo do seu desenvolvimento.

O NetOffice possui várias funções importantes que podem ser usadas pelo gerente do projeto ao longo de sua duração. Inicialmente, a ferramenta possui gerenciamento de usuários, níveis de acesso com permissões configuráveis, cadastro e monitoramento de tarefas, controle de arquivos de aprovação, notas, CRM, gráficos de Gantt, relatórios, entre outros recursos (NETOFFICE, 2011).

## **2.4 Modelagem de Dados**

Segundo Teorey et al. (2006), a modelagem de dados é o principal ponto de um projeto lógico de banco de dados. A modelagem de dados deve basear na simplicidade e na legibilidade, pois o objetivo é criar uma estrutura de dados baseado em uma situação do mundo real, sendo que essa estrutura deve ser simples e significativa para seus usuários.

### **2.4.1 Modelo Conceitual**

Segundo Elmasri e Navathe (2011), modelos conceituais ou de alto nível possuem conceitos que descrevem os dados como os usuários os percebem.

O modelo conceitual pode ser considerado como primeiro passo de um projeto de banco de dados, pois ele representa a realidade do problema de uma

forma mais geral, independentemente de restrições de implementação. O principal foco nesse ponto é constituir uma ideia dos dados e relacionamentos da situação em questão. O resultado obtido do modelo conceitual é semelhante a um esquema que representa as informações reais, bem como as estruturas de dados para representar essas informações (MACHADO et al., 2009). Um exemplo de estrutura utilizada nessa fase é o modelo entidade-relacionamento e o modelo UML.

#### 2.4.1.1 Modelo Entidade-Relacionamento

O modelo entidade-relacionamento (MER) foi definido em 1976, por Peten Chen, baseado na teoria relacional de E. F. Codd, de 1970. Atualmente, o modelo entidade-relacionamento foi incrementado e expandido por diversos autores.

De modo geral, o modelo entidade-relacionamento é utilizado como base para um bom entendimento do usuário perante o banco de dados conceitual. Basicamente, o modelo entidade-relacionamento é constituído de três classes de objetos: entidades, relacionamentos e atributos (TEOREY et al., 2006).

Nadeau et al. (2006) define cada uma dessas três classes de objetos, conforme a Tabela 1.

Tabela 1 Classes de objetos de um modelo entidade-relacionamento

Classes de Objetos	Descrição
<b>Entidades</b>	As entidades podem ser definidas como o principal ponto de onde as informações são coletadas. Podem ser representadas por qualquer evento, objeto, pessoa, etc. Sua representação se dá através de um retângulo.

<b>Relacionamentos</b>	Os relacionamentos realizam associações entre entidades. Eles são descritos em três tipos de termos: grau, conectividade e existência.
<b>Atributos</b>	Os atributos podem ser definidos como os detalhes que descrevem as entidades. Um atributo pode ter seu valor dentro de uma entidade. São classificados em dois tipos: identificadores e descritores. Identificadores são aqueles utilizados para determinar exclusividade, ao contrário dos descritores.

De acordo com Machado et al. (2009), o grau de relacionamento entre duas entidades é definido pelo número de ocorrências entre ambas. São três graus de relacionamentos distintos:

1. Relacionamento de Um-para-Um: cada elemento de uma entidade relaciona-se com um e somente um elemento de outra entidade.
2. Relacionamento de Um-para-Muitos: a ideia é que um elemento da primeira entidade se relaciona com muitos da segunda entidade, mas para cada elemento da segunda entidade deve-se haver apenas um elemento da segunda entidade.
3. Relacionamento de Muitos-para-Muitos: um elemento da primeira entidade se relaciona com muitos da segunda entidade e, também, cada elemento da segunda entidade pode se relacionar com vários elementos da segunda entidade.

Ainda segundo Machado et al. (2009), o grau de um relacionamento pode ultrapassar de duas entidades. Os relacionamentos entre múltiplas entidades se dão quando todas as ocorrências do relacionamento possuem relação com todas as entidades envolvidas no relacionamento.

O modelo entidade-relacionamento, apesar de lógico, é a base para a criação do modelo relacional, próxima etapa para a modelagem de um banco de dados.

#### **2.4.1.2 UML**

Segundo Fowler (2002), UML (*Unified Modeling Language*) é uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de software. O mais comum dos casos é utilizar a UML como uma espécie de esboço, pois nessa utilização, a UML ajuda a transmitir alguns aspectos de um sistema, ou seja, tem o objetivo de explicitar as ideias e alternativas sobre o que está prestes a fazer.

A UML tem por maior utilização a representação gráfica de projetos de linguagem de programação orientada a objetos, porém, sua utilização em banco de dados relacionais tem sido bastante utilizada. Segundo Correia (2002), a versatilidade do paradigma orientado a objetos permite o projeto não só de sistemas e programas, mas também de base de dados.

Para utilização em base de dados relacionais, a UML precisa, segundo Correia (2002), adicionar detalhes que não existem no modelo de objetos, como por exemplo, notações de chaves primárias e estrangeiras. A idéia é representar a tabela como sendo uma classe e seus atributos, podendo relacionar as tabelas entre si como meio de associação, através de chaves estrangeiras e relação.

### **2.4.2 Modelo Relacional**

Pode ser denominado de modelo intermediário, representacional ou de implementação, segundo Elmasri e Navathe (2011).

Baseado no modelo conceitual, o modelo relacional passa a descrever as estruturas que serão criadas no banco de dados. Nesse ponto deve haver uma abordagem, mesmo que superficial, para as possibilidades permitidas pela aplicação a ser construída.

Esse modelo também oferece conceitos que podem ser entendidos pelos usuários finais, mas não excessivamente distantes da forma como os dados estão organizados no computador.

### **2.4.3 Modelo Físico**

Segundo Abreu et al. (2009), o modelo físico apresenta as estruturas físicas de armazenamento de dados descritas no modelo relacional. Nesse ponto acontece a criação das tabelas com seus campos, relacionamentos, índices, etc. Nessa fase é interessante haver um levantamento dos futuros impactos de cada tipo de implementação para a necessidade a ser satisfeita.

O modelo físico de dados contem conceitos que descrevem detalhes de como os dados estão armazenados no computador (ELMASRI; NAVATHE, 2011).

## **2.5 Banco de Dados Relacional**

Segundo Machado et al. (2010), o homem procura transmitir e documentar informações desde a antiguidade. Atualmente, com o advento dos Sistemas Gerenciadores de Banco de Dados (SGBDs), as informações são

armazenadas computacionalmente, ou seja, guardadas em uma base de dados, sendo elas acessadas quando necessário. Segundo Date (2003), um banco de dados pode ser definido como um sistema computacional capaz de armazenar e processar dados e, quando solicitado, disponibilizar esses dados ao usuário.

Existem vários modelos de dados, porém o mais comum e utilizado deles é o modelo relacional, até para meios didáticos também.

Segundo Elmasri e Navathe (2011), um modelo relacional representa o banco de dados como um conjunto de relações, sendo essas parecidas a uma tabela de valores, onde cada registro tem uma estrutura linear ou plana. Definindo formalmente, uma linha é chamada de tupla, uma coluna é denominada de atributo e a tabela é chamada de relação. Na Figura 2, tem-se um exemplo de uma tabela de um modelo relacional.

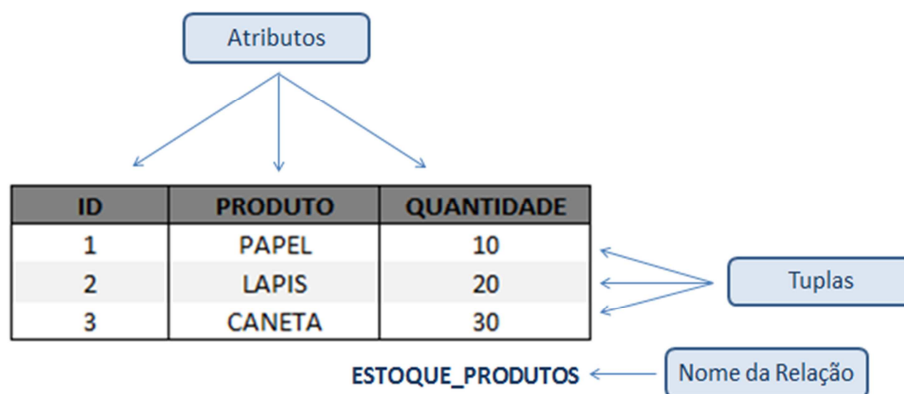


Figura 2 Representação de uma tabela no modelo dados relacional

A Figura 2 mostra o exemplo de uma tabela onde, o nome da relação é ESTOQUE\_PRODUTOS, tendo três registros armazenados, denominados de tuplas, sendo essas caracterizadas por três campos, chamados de atributos, que são ID, PRODUTO e QUANTIDADE. Traduzindo os dados em informações, percebe-se que o estoque de um determinado lugar possui dez folhas de papel,

vinte unidades de lápis e 30 unidades de canetas, sendo que os produtos são identificados por códigos.

Um Sistema Gerenciador de Banco de Dados pode ser definido como uma coleção de dados relacionados que são acessados por usuários e aplicativos. A coleção de dados constitui o banco de dados, que pode conter informações de grande relevância para a sua utilização. De modo geral, o objetivo de um Sistema Gerenciador de Banco de Dados é garantir a integridade dos dados e disponibilizá-los de maneira eficiente (SILBERSCHATZ et al., 2006).

Segundo Teorey et al. (2007), os sistemas de banco de dados evoluíram ultimamente à medida que aplicações comerciais têm sido dominadas pelo modelo de dados relacional, pois ele permite focalizar o projeto lógico e o projeto físico.

Os dados são considerados como de fundamental importância nos processos das mais diversas áreas de atuação. Para a gerência de projetos de *software*, os dados de reuniões, tarefas, custos, prazos e outros são armazenados para se manter um controle e planejamento dos projetos. Nesse contexto, a modelagem de banco de dados é fator fundamental para projetos de TI, fazendo com que sua importância aumentasse significativamente.



### **3 METODOLOGIA**

Este capítulo descreve a metodologia utilizada no trabalho para que possam ser alcançados os objetivos propostos pela pesquisa. Na primeira seção é apresentada a classificação da pesquisa quanto à natureza, ao objetivo e aos procedimentos. Em seguida são descritos os procedimentos metodológicos da pesquisa.

#### **3.1 Tipo da Pesquisa**

O trabalho foi realizado perante uma pesquisa qualitativa. Segundo Wainer (2007), a pesquisa qualitativa é caracterizada pela observação dos ambientes onde o sistema está sendo utilizado e também do entendimento das várias perspectivas dos usuários do sistema.

A pesquisa qualitativa foi utilizada para verificar questões básicas e superficiais do funcionamento de um ambiente *online* de gerência de projetos. Para isso, foi feita uma observação sobre as principais funcionalidades de três ferramentas muito utilizadas atualmente: dotProject, ClokingIt e NetOffice.

A observação dos ambientes em si foi feita de maneira descritiva para familiarização com o contexto e o funcionamento dos ambientes de Gerência de Projetos. Baseado nessa observação foi criado um banco de dados para dar suporte à carga de informações a serem processadas e armazenadas em um ambiente de Gerência de Projetos a ser desenvolvido futuramente.

#### **3.2 Procedimentos Metodológicos**

O desenvolvimento do trabalho ocorreu três etapas principais. A primeira delas foi a análise dos três ambientes de gerência de projetos,

DotProject, ClockingIt e NetOffice, onde em seguida foi descrito para cada ambiente as características e informações de cada um deles. Nessa etapa, houve uma aquisição de conhecimento acerca dos ambientes de gerência de projetos, baseado na observação e descrição deles.

Com a contextualização necessária adquirida, a segunda etapa consistiu em definir quais funcionalidades e características deveria ter o novo banco de dados. Foram analisadas características eficientes ou não entre os ambientes.

A terceira etapa consistiu na modelagem do banco de dados. O modelo conceitual foi o primeiro passo, nele foi elaborado um rascunho para as principais ideias da do novo banco de dados. O segundo passo foi a construção do modelo relacional, onde foi utilizada a ferramenta MySQL Workbench para definição das estruturas e relacionamentos das tabelas do banco de dados. A terceira e última etapa, foi a criação da estrutura física, onde foi utilizado o Sistema Gerenciador de Banco de Dados MySQL para teste do novo banco de dados criado.

### **3.3 Ferramentas Utilizadas**

O trabalho foi desenvolvido com auxílio de algumas ferramentas. Para análise dos ambientes de projeto, foi utilizado os ambientes de gerência de projetos DotProject, NetOffice e ClockingIt. Os dois primeiros foram instalados e configurados em uma máquina local. O último foi utilizado no próprio ambiente disponibilizado pela ClockingIt, onde é possível criar um domínio de uso particular.

Para criação do modelo de banco de dados e criação de sua estrutura física, foi utilizado o software gratuito MySQL Workbench versão 5.2.34 CE - Revision 7780.

## 4 ANÁLISE DOS AMBIENTES DOTPROJECT, CLOCKINGIT E NETOFFICE

Considerar os ambientes de gerência de projetos já consolidados no mercado é muito importante para entender as principais características que esse tipo de aplicativo deve ter.

Parte agora uma análise das funcionalidades de cada ambiente com o objetivo de levantar pontos fundamentais para serem utilizados na nova estrutura de banco de dados.

### 4.1 Análise do DotProject

O ambiente do DotProject permite ao usuário uma série de cadastros, sejam eles de projetos, tarefas, clientes, arquivos, contatos, fóruns, dentre outros. A estrutura de banco de dados do DotProject é composta de 68 tabelas que dão suporte a gravação de dados de todo o sistema.

O DotProject é subdivido em módulos, os quais são utilizados diretamente pelos envolvidos nos projetos. Na instalação inicial, 12 módulos são automaticamente ativados, e outros 5 são opcionais, totalizando 17 módulos disponíveis no sistema.

Segue agora uma pequena descrição de cada funcionalidade do ambiente. A versão utilizada para análise foi a 2.1.5.

- **Multiusuário:** permite a utilização simultânea de vários usuários. Quando o ambiente é inicializado, aparece para o usuário uma tela de login para ele se autenticar, conforme a Figura 3.

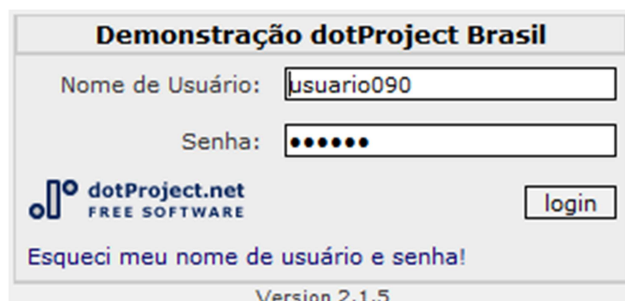


Figura 3 Tela de login dotProject

- **Sistema de permissões:** Quando se cria um usuário, existe a possibilidade de conceder permissões a usuários específicos.

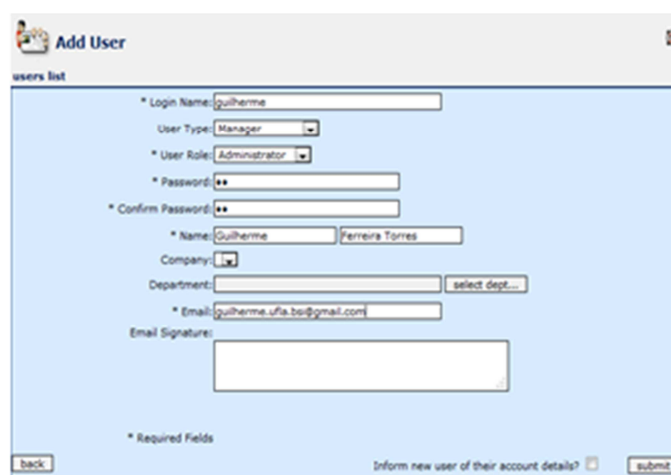


Figura 4 Tela de cadastro de usuário dotProject

A Figura 4 mostra a opção de definir o tipo de usuário e também a sua permissão do sistema. O DotProject permite a criação de níveis de permissões, a qual consiste em um conjunto de permissões dadas a determinado nível baseado nos módulos do sistema.

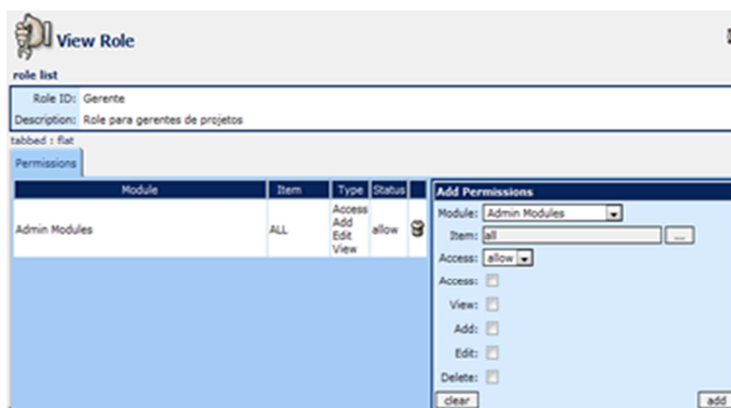


Figura 5 Tela de edição de role dotProject

A Figura 5 demonstra a criação de uma role para Gerentes de Projetos. Nesse caso, o nível de permissão possui acesso para todos os módulos administrativos.

- **Gerenciamento de empresas:** o ambiente permite o cadastro de companhias, sendo elas subdivididas em grupos específicos, tais como, clientes, fornecedores, consultores, governamentais, internos e outros.

Figura 6 Tela de cadastro de companhia do dotProject

A Figura 6 demonstra o exemplo de criação de uma empresa, onde é permitido adicionar os dados da empresa em questão e também o seu tipo, que nesse caso se trata de um cliente.

- **Cadastro de Projetos:** permite o cadastro de projetos para atender demandas internas ou externas.

The screenshot shows a web form titled "New Project" with a "projects list" header. The form contains several input fields and dropdown menus. The "Project Name" is "POI1-001-Implantação ABC", "Project Owner" is "Ferreira Torres, Guilherme", and "Company" is "ABC Transportes Rodoviários". The "Start Date" is "03/04/2012" and "Target Finish Date" is "25/05/2012". The "Target Budget R\$" is "5000,00". The "Priority" is "high", "Short Name" is "POI1-001", and "Color Identifier" is "FFFFFF". The "Project Type" is "Operative" and "Status" is "Proposed". The "Progress" is "0,0%". The "Description" is "Projeto para implantação de sistema para gerenciar frota da ABC Transportes". There are "cancel" and "submit" buttons at the top and bottom of the form.

Figura 7 Tela de cadastro de projeto do dotProject

Na Figura 7, tem-se um exemplo de cadastro de projeto, onde é possível detalhar as características de cada projeto, tais como, responsável pelo projeto, cliente, status, custos, datas previstas, entre outros.

- **Registro de atividades:** permite o cadastro de atividades para os projetos existentes.

**Add Task**

tasks list : view this project

Project: POII-001-Implantação ABC

Task Name \*  
Levantamento de requisitos

Status: Active | Priority \*: high

Progress: 0 % | Milestone?

\* indicates required field cancel save

tabbed : flat

Details | Dates | Dependencies | Human Resources

Task Owner: Ferreira Torres, Guilherme | Task Type: Operative

Access: Public | Select contacts...

Web Address:

Task Parent: None | Target Budget: R\$ 250,00

Description:  
Tarefa para levantamento de requisitos no cliente

\* indicates required field cancel save

Figura 8 Tela de cadastro de atividades do dotProject

A Figura 8 mostra o exemplo de um cadastro de atividade para um determinado projeto. No cadastro da atividade é possível colocar o responsável pela tarefa, o tipo de tarefa, seu tipo de acesso, prioridade, status, descrição, datas, dependências, progresso, entre outros. Para o usuário realizar o lançamento de horas nessa atividade, ele deve acessar o link da atividade, lançar um total de tempo trabalhado e atualizar o percentual de concretização da tarefa.

- **Repositório de documentos:** permite o armazenamento de arquivos a serem utilizados pelos usuários.

**Add File**

files list

Folder: Root

Version: 1

Category: Document

Project: POII-001-Implantação ABC

Task: select task...

Description: Memorial

Upload File: C:\Users\Guilherme\Doc Selecionar arquivo...

Final Version

Notify Assignees of Task or Project Owner by Email

Notify Project and Task Contacts

cancel submit

Figura 9 Tela de cadastro de arquivos do dotProject

A Figura 9 mostra a tela de adição de arquivos do DotProject, onde é possível escolher um arquivo disponível na máquina do usuário e disponibilizar o mesmo no ambiente de acordo com o seu tipo, projeto ou até mesmo atividade.

- **Calendário integrado:** pode ser utilizado para verificar as alocações e compromissos dos usuários.

**Monthly Calendar**

maio 2012

seg	ter	qua	qui	sex
	1 Levantamento de requ...	2	3 Levantamento de requ...	4
7	8	9	10	11

Figura 10 Parte do calendário de alocações do dotProject



A Figura 10 mostra um exemplo de alocação de um funcionário do dia 01/05/2012 até 03/05/2012, onde ele está alocado para a tarefa Levantamento de Requisitos. É possível também clicar sobre a alocação que o ambiente redireciona o usuário para a página da atividade em questão.

- **Fórum de discussão:** podem ser criados fóruns para interação dos usuários.

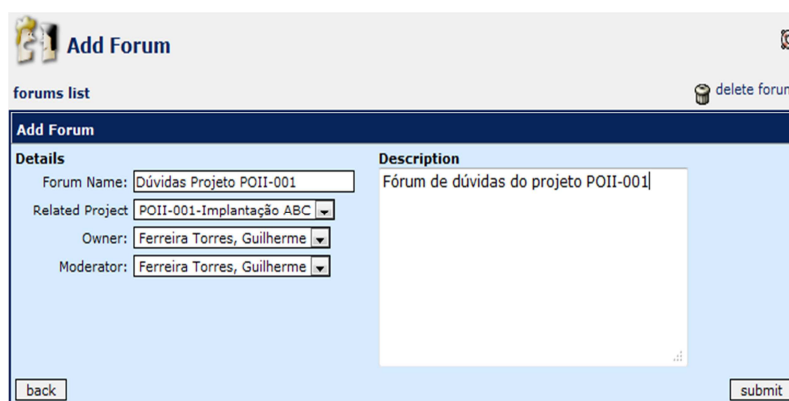
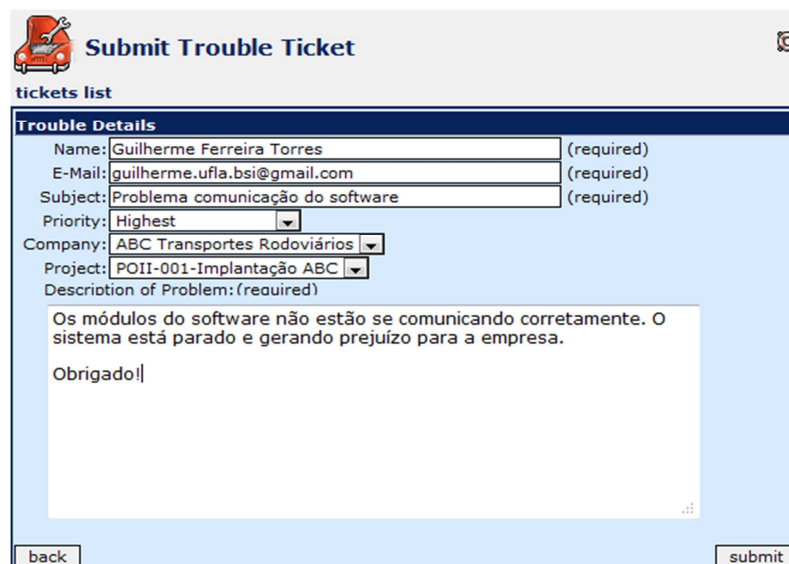


Figura 11 Criação de fórum no dotProject

A Figura 11 é um exemplo de criação de fórum para determinado projeto. Além disso, é possível adicionar um dono e um moderador para o fórum.

- **Chamada de suporte:** a funcionalidade permite aos usuários o cadastro de chamadas de suporte pelos clientes.



**Submit Trouble Ticket**

tickets list

**Trouble Details**

Name:  (required)

E-Mail:  (required)

Subject:  (required)

Priority:

Company:

Project:

Description of Problem: (required)

Os módulos do software não estão se comunicando corretamente. O sistema está parado e gerando prejuízo para a empresa.

Obrigado!|

Figura 12 Criação de chamada de serviço do cliente no dotProject

O dotProject permite ao usuário, seja ele cliente ou não, cadastrar chamadas de suporte relatando possíveis erros do sistema ou serviço. No exemplo da Figura 12, tem-se uma criação de chamada de suporte, onde foi detalhado projeto e qual cliente que fez a chamada. Após o cadastro, a chamada aparece para o responsável pelo projeto.

Como foco geral, o dotProject visa o planejamento dos projetos, sendo o gerente de projetos o maior utilizador do sistema, tendo ele a oportunidade de cadastrar projetos, atividades, recursos, entre outros. As equipes de desenvolvedores dos projetos podem usar o ambiente para verificar e atualizar suas alocações e atividades, bem como utilizar de fóruns e acesso a arquivos disponíveis.

Como visto, o ambiente possui muitas funcionalidades, porém a disponibilização de muitas delas de maneira distintas pode gerar uma confusão

ao usuário na hora de utilizar o ambiente, prejudicando um pouco sua usabilidade.

Quanto suas funcionalidades, é notada a falta de uma que trate os custos do projeto, ponto esse de extrema importância para o gerente de projetos controlar seus gastos.

## 4.2 Análise do ClockingIt

O ClockingIt é um ambiente *online* muito intuitivo. Para um ambiente de gerência de projetos, ele se limita a criação de projetos, tarefas, lançamento de horas da equipe em suas tarefas e relatórios personalizados que o usuário pode montar. Além disso, a ferramenta possui outras funcionalidades, tais como wiki integrada, fóruns, bate-papo, arquivos, entre outros.

Segue então a descrição das principais funcionalidades do ClockingIt:

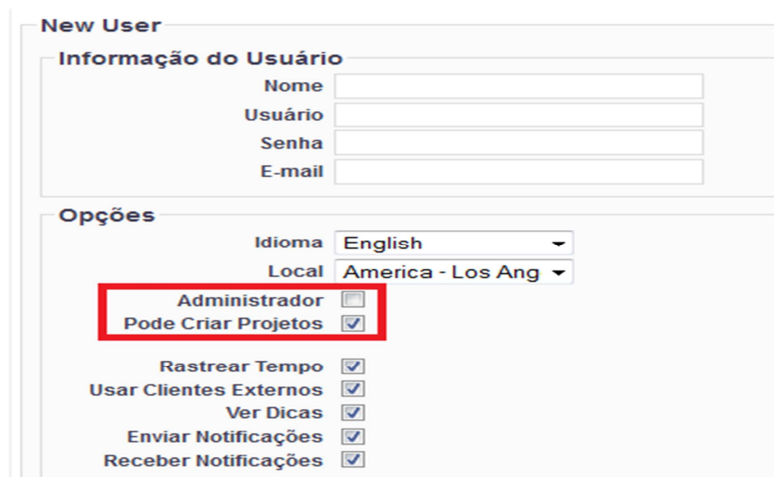
- **Multiusuário:** permite a utilização simultânea de vários usuários.



Figura 13 Tela inicial de login do ClockingIt

A Figura 13 mostra a tela de login do ambiente ClockingIt, onde o usuário deve realizar a autenticação para utilização do ambiente.

- **Sistema de permissões:** possibilidade de conceder diferentes permissões a usuários específicos.



**New User**

**Informação do Usuário**

Nome

Usuário

Senha

E-mail

**Opções**

Idioma

Local

Administrador

Pode Criar Projetos

Rastrear Tempo

Usar Clientes Externos

Ver Dicas

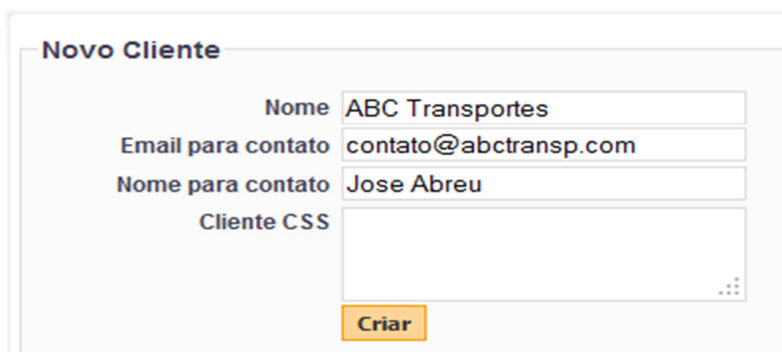
Enviar Notificações

Receber Notificações

Figura 14 Tela de criação de usuário do ClockingIt

Conforme mostra a Figura 14, o ClockingIt possui um sistema de permissão, onde ao inserir um usuário ele tem a permissão de ser administrador do ambiente, poder criar projetos ou ser apenas um usuário comum.

- **Gerenciamento de clientes:** capacidade de cadastro de clientes.



**Novo Cliente**

Nome

Email para contato

Nome para contato

Cliente CSS

Figura 15 Tela de adição de cliente do ClockingIt

A Figura 15 é um exemplo de criação de cliente do ClockingIt, onde é possível adicionar o e-mail de contato e o responsável pelo cliente.

- **Cadastro de Projetos:** permite o cadastro de projetos para atender demandas internas ou externas.

**Novo projeto**

Nome: POII-001-Implantação\_Software

Cliente: ABC Transportes

Descrição: Implantação de software na ABC Transportes

Criar Fórum:

**Criar** e copiar permissões de: [Nenhum]

Figura 16 Tela de adição de projeto do ClockingIt

A Figura 16 mostra o exemplo da criação de um projeto, onde é possível colocar o cliente e a descrição do projeto. Além disso, o ClockingIt permite a criação de metas de projetos, onde é estipulado um prazo pra essa meta ser cumprida, conforme mostra o exemplo da Figura 17.

**Nova meta**

Nome: Não ultrapassar o orçamento

Prazo: 31/05/2012

Proprietário: Guilherme Torres

Descrição: Meta para não ultrapassar o orçamento inicial previsto para o projeto.

**Criar**

Figura 17 Tela de adição de meta de projeto do ClockingIt

Ainda dentro de projetos, o ambiente disponibiliza uma tela para edição de permissão de usuários dentro do projeto.

Acesso ao Projeto												
Conectado	Usuário	Ler	Comentário / nova interação	Trabalho	Fechar	Criar	Editar	Alocar	Priorizar	Metas	Relatórios	Administrar Todos
	Guilherme Torres											
	Joaquim da Silva	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 18 Tela de edição de permissão de projeto do ClockingIt

O ClockingIt suporta a distribuição de permissões específicas aos usuários dentro de um projeto qualquer. No exemplo da Figura 18, dois usuários possuem permissões diferentes para um mesmo projeto.

- **Registro de atividades:** permite o cadastro de atividades para os projetos existentes.

**NOVA TAREFA**

**Informação**

Resumo: Levantamento de Requisitos

Tags:

Descrição: Realizar o levantamento de requisitos no cliente para o projeto.

Anexar arquivo: C:\Users\Guilherme\Documents\Doc\_Le... Selecionar arquivo...

Enviar notificações por e-mail

**Alvo**

Projeto: POII-001-Implantaç

Meta: [Nenhum]

Alocado para: Joaozinho da Silva

Requisitado por:

**Atributos**

Tipo:  Tarefa  Nova característica  Defeito  Melhoria

Prioridade: Alta

Importância: Principal

Tempo estimado: 20

Prazo: 09/05/2012

**Dependências**

Gera dependência em #

**Notificação**

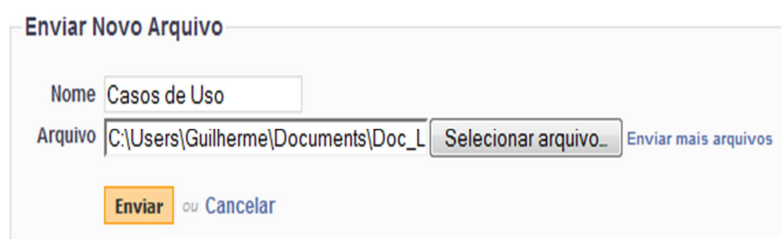
Nome / Email

Figura 19 Tela de criação de atividades de projeto do ClockingIt

Na Figura 19, tem-se um exemplo de criação de atividade no ClockingIt, onde o usuário pode anexar um arquivo a atividade, escolher o projeto a qual a atividade pertence, definir qual usuário vai realizar a atividade, definir a prioridade e importância, o seu tempo estimado e prazo, bem como se existe alguma dependência pra essa atividade ou não.

Uma observação importante é que o ClockingIt trata o tipo de atividade cadastrada, podendo ser:

- Tarefa: geralmente utilizada para atividades de projetos;
  - Nova Característica: alguma nova característica do projeto em questão. Pode ser utilizada para alertar sobre mudança de escopo, entre outros;
  - Defeito: pode ser utilizado para tratar falhas do projeto em desenvolvimento;
  - Melhoria: utilizado para indicar melhorias no projeto.
- **Repositório de arquivos:** permite o armazenamento de arquivos a serem utilizados pelos usuários.



Enviar Novo Arquivo

Nome Casos de Uso

Arquivo C:\Users\Guilherme\Documents\Doc\_L Selecionar arquivo... Enviar mais arquivos

Enviar ou Cancelar

Figura 20 Tela de envio de arquivos do ClockingIt

A Figura 20 mostra o exemplo de envio de um arquivo para o ambiente ClockingIt. Além do envio de arquivos, é permitido a criação de diretórios para armazenar esses arquivos, tendo os diretórios ligação direta com um

determinado projeto. Dessa forma apenas os usuários dos projetos tem acesso aos arquivos do diretório em questão.

- **Calendário integrado:** pode ser utilizado para verificar as alocações e compromissos dos usuários.

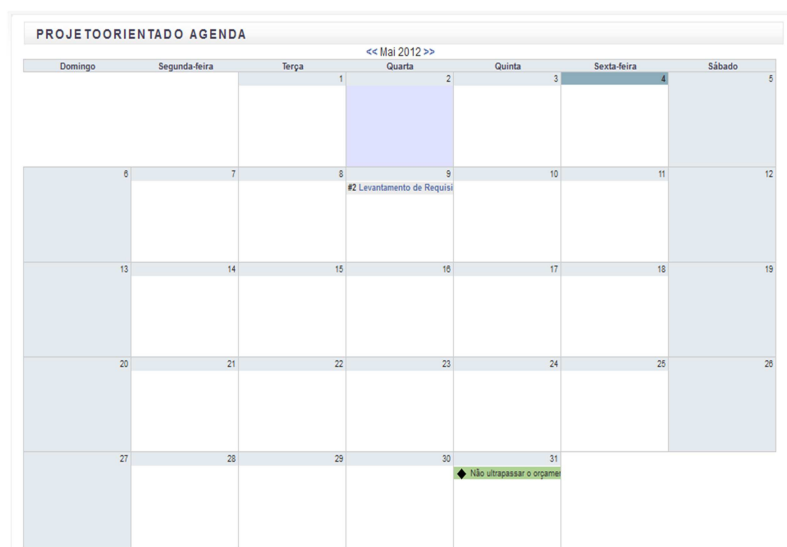
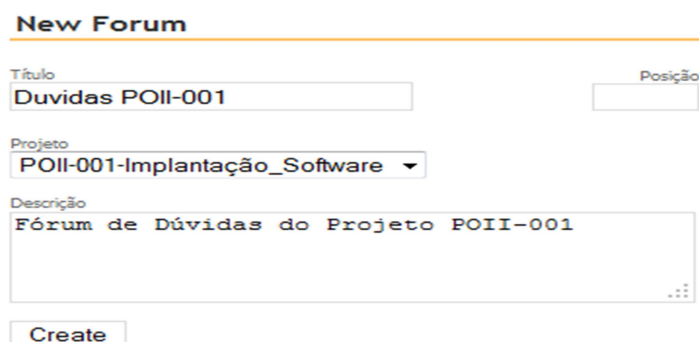


Figura 21 Tela calendário do ClockingIt

A Figura 21 mostra o calendário integrado de um determinado usuário do ClockingIt. O calendário do ClockingIt mostra todas as alocações do usuário em seus projetos, onde é possível o usuário clicar na atividade e ser redirecionado pra página daquela atividade.

- **Fórum de discussão:** podem ser criados fóruns para interação dos usuários.





**New Forum**

Título: Duvidas POII-001

Posição:

Projeto: POII-001-Implantação\_Software

Descrição: Fórum de Dúvidas do Projeto POII-001

Create

Figura 22 Tela de criação de fóruns do ClockingIt

Na Figura 22, tem-se o exemplo de criação de um fórum para dúvidas de um projeto. Quando se cria um fórum, existe a possibilidade de especificar esse fórum a determinado projeto ou então deixar ele livre para todos os usuários utilizarem. Dentro do fórum, é possível criar tópicos e respondê-los.

- **Enciclopédia integrada:** permite a inserção de páginas web no formato de enciclopédias wiki.



Figura 23 Tela principal da wiki integrada do ClockingIt

A Figura 23 mostra a tela inicial da Wiki integrada do ClockingIt. Nela é permitida a inserção de imagens, tabelas, links, páginas web, entre outros. Qualquer usuário pode editar o seu conteúdo.

- **Relatórios:** permite a criação de relatórios customizados.



**RELATÓRIOS**

**Configurar relatório**

Tipo de relatório: Carga de trabalho ▼

Linhas: Tarefas ▼

Colunas: Data ▼

**Filtro**

Cliente: ABC Transportes ▼

Projeto: POII-001-Implantaç ▼

Usuário: [Qualquer Usuário] ▼

[Opções Avançadas](#)

**Executar Relatório**

Figura 24 Tela de criação de relatório do ClockingIt

A Figura 24 demonstra o exemplo de criação de um relatório personalizado do ClockingIt. Existem quatro tipos de relatórios para serem escolhidos. Ainda é possível configurar outros atributos, tais como cliente, projeto, usuário, entre outros.

- **Ferramenta de comunicação:** permite criar salas de bate-papo para os usuários.



**SALAS DE BATE-PAPO**

**Nova Sala**

Nome: Bate Papo POII-001

Projeto: POII-001-Implantaç

Descrição: Sala de bate papo do projeto POII-001

**Criar** ou **Cancelar**

Figura 25 Tela de criação de bate-papo do ClockingIt

A Figura 25 mostra um exemplo de criação de uma sala de papo-papo para um determinado projeto. As salas de bate-papo podem ser restritas a um projeto ou públicas entre todos os usuários do ClockingIt.

De uma maneira geral, o ClockingIt é uma ferramenta que proporciona para o usuário a oportunidade trabalhar com a gerência de projetos em si, além de possuir outros importantes módulos, tais como fóruns, bate-papo, enciclopédia wiki e armazenamento de arquivos.

O ambiente é um pouco econômico no conteúdo de suas funcionalidades, pois muitas delas são demasiadamente simples. O gerente de projetos pode ter dificuldade em realizar algumas atividades mais específicas na ferramenta.

Para utilização dos pontos mais simples da gerência de projetos, mostra ser uma boa ferramenta com uma boa interface.

### 4.3 Análise do NetOffice

O ambiente NetOffice permite ao usuário realizar cadastros de projetos, tarefas, clientes, bookmarks, setores, lançamento de horas, entre outros. A estrutura de banco de dados do NetOffice é composta de 29 tabelas que dão

suporte a gravação de dados de todo o sistema. Abaixo, segue algumas de suas principais funcionalidades:

- **Multiusuário:** permite a utilização simultânea de vários usuários.

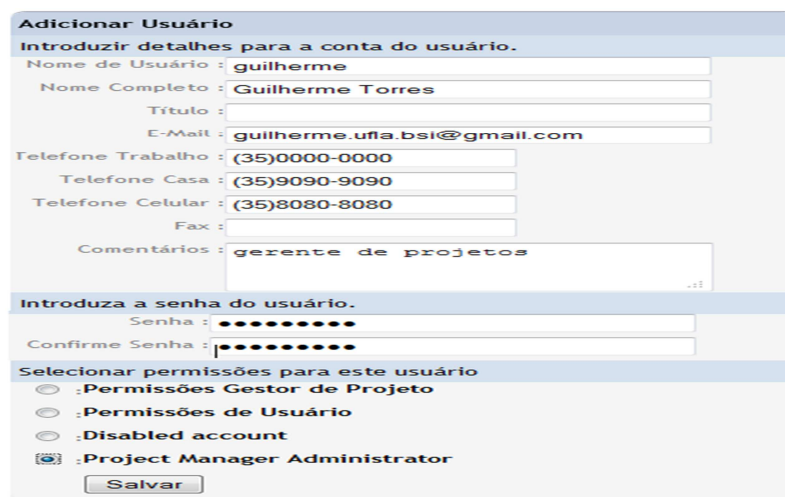


A screenshot of the NetOffice login screen. The window title is "NetOffice : Identificação". It features a language dropdown menu set to "Brazilian Portuguese (Default)". There are two input fields: "\* Nome de Usuário" containing "guilherme" and "\* Senha" with masked characters. A "Lembrar da senha" checkbox is present and unchecked. A "Identificação" button is centered below the fields. At the bottom, there is a blue link that says "Esqueceu a senha ?".

Figura 26 Tela de login do NetOffice

A Figura 26 representa a tela de login do NetOffice, onde o usuário deve se autenticar para utilização do ambiente.

- **Sistema de permissões:** possibilidade de conceder permissões a usuários específicos.

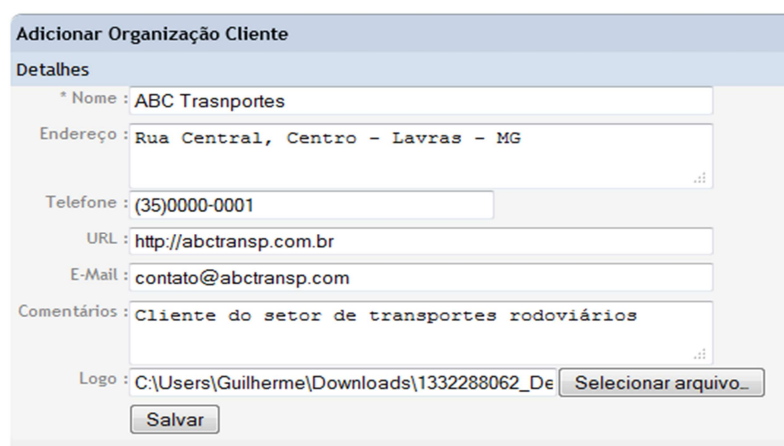


A screenshot of the NetOffice user registration screen. The window title is "Adicionar Usuário". It is divided into three sections. The first section, "Introduzir detalhes para a conta do usuário.", contains fields for "Nome de Usuário" (guilherme), "Nome Completo" (Guilherme Torres), "Título", "E-Mail" (guilherme.ufia.bsl@gmail.com), "Telefone Trabalho" ((35)0000-0000), "Telefone Casa" ((35)9090-9090), "Telefone Celular" ((35)8080-8080), "Fax", and "Comentários" (gerente de projetos). The second section, "Introduza a senha do usuário.", has "Senha" and "Confirme Senha" fields, both masked. The third section, "Selecionar permissões para este usuário", has radio buttons for ".Permissões Gestor de Projeto", ".Permissões de Usuário", ".Disabled account", and ".Project Manager Administrator" (which is selected). A "Salvar" button is at the bottom.

Figura 27 Tela de cadastro de usuário do NetOffice

A Figura 27 mostra o exemplo de criação de um usuário. Ao criar o usuário, deve-se escolher o seu nível de permissão no ambiente, que é subdividido em três permissões (Gestor de Projeto, Usuário e *Project Manager Administrator*), além de outro tipo que é de conta desativada.

- **Gerenciamento de clientes:** capacidade de cadastro de clientes dos projetos.



Adicionar Organização Cliente

Detalhes

\* Nome : ABC Trasportes

Endereço : Rua Central, Centro - Lavras - MG

Telefone : (35)0000-0001

URL : http://abctransp.com.br

E-Mail : contato@abctransp.com

Comentários : Cliente do setor de transportes rodoviários

Logo : C:\Users\Guilherme\Downloads\1332288062\_De Selecionar arquivo...

Salvar

Figura 28 Tela de cadastro de clientes do NetOffice

Na Figura 28, tem-se um exemplo de criação de um cliente. O NetOffice permite cadastrar o cliente com algumas características, tais como endereço, telefone, URL, email, comentários e um logo.

- **Cadastro de Projetos:** permite o cadastro de projetos para atender demandas internas ou externas.

The screenshot shows a web form titled "Adicionar Projeto" with a sub-section "Detalhes". The form contains the following fields and values:

- Nome: POII-001-Implantação\_software
- Prioridade: Média (dropdown menu)
- Descrição: Implantação de software na ABC Transportes
- URL do Site em Desenvolvimento: (empty text box)
- URL do Site de Produção: (empty text box)
- Dono: guilherme / Guilherme Torres (dropdown menu)
- Organização Cliente: ABC Trasnportes (dropdown menu)
- Habilitar Fases: Application (dropdown menu)
- Estado: Aberto (dropdown menu)
- Tipo: External project (dropdown menu)
- Tamanho máximo de arquivo: 51200 Bytes

A "Salvar" button is located at the bottom of the form.

Figura 29 Tela de cadastro de projetos do NetOffice

A Figura 29 mostra o exemplo de criação de um projeto no NetOffice, onde é possível definir prioridade do projeto, bem como seu estado, cliente, tipo, descrição, líder, entre outros. Além disso, o NetOffice subdivide o projetos em seis fases (Concepção, Planejamento, Desenvolvimento, Teste, Rollout e Manutenção). Ao se criar uma tarefa, o usuário deve definir a fase correspondente da mesma.

- **Registro de tarefas:** permite o cadastro de tarefas para os projetos existentes.

**Adicionar Tarefas**

**Info**

Projeto : POII-001-Implantação\_software ▾

Fase : [Concept](#)

Organização Cliente : ABC Trasnportes

**Detalhes**

Nome : Levantamento de requisitos

Descrição : Levantar requisitos junto ao cliente

Delegado a : guilherme / Guilherme Torres ▾

Milestone :

Fase : Concept ▾

Estado : Aberto ▾

Finalização : 0 % ▾

Prioridade : Média ▾

Data de início : 2012-05-05 ...

Data de Finalização : -- ...

Tempo Estimado : 30 horas

Service : None ▾

Comentários :

Publicado :

Salvar

Figura 30 Tela de cadastro de atividades do NetOffice

Na Figura 30, tem-se um exemplo de criação de uma tarefa de um projeto no NetOffice. É possível adicionar descrição da tarefa, sua fase no projeto, o estado da tarefa, prioridade, datas, tempo estimado, entre outros.

- **Calendário de alocações:** pode ser utilizado para verificar as alocações dos usuários, baseado nas tarefas delegadas a cada um.

Maio 2012						
Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
		1	2	3	4	5
					Tarefa: <a href="#">Levantamento de requisitos</a> (Data de início)	6
	7	8	9	10	11	12
	<a href="#">14</a>	<a href="#">15</a>	<a href="#">16</a>	<a href="#">17</a>	<a href="#">18</a>	<a href="#">19</a>
	21	22	23	24	25	26
	28	29	30	31		

Figura 31 Calendário integrado do NetOffice

O calendário da Figura 31 demonstra a alocação de um usuário, onde este tem uma tarefa iniciada no dia 05. Ao clicar sobre a alocação, o ambiente direciona o usuário à página da atividade.

- **Relatórios:** permite a criação de relatórios customizados.

**Criar Relatório**

Clientes : [Selecionar Todos](#)  
ABC Transportes

Projetos : [Selecionar Todos](#)  
POII-001-Implantação\_software

Delegado a : [Selecionar Todos](#)  
Não Delegada  
admin  
carlos

Data de Finalização :  Todas as datas  
 Entre as datas

Complete date :  Todas as datas  
 Entre as datas

Estado : [Selecionar Todos](#)  
Finalizado pelo Cliente  
Finalizado  
Não Iniciado

Prioridade : [Selecionar Todos](#)  
Nenhuma  
Muito Baixa  
Baixa

Figura 32 Criação de relatórios do NetOffice



A Figura 32 mostra o exemplo de criação de um relatório customizado, onde é permitido filtrar clientes, projetos, delegações de tarefas, datas, estados e prioridades. O NetOffice permite também salvar os relatórios criados.

O NetOffice mostra ser uma boa ferramenta para o usuário, principalmente no gerenciamento de projetos, pois ao criar um projeto, ele já é subdividido em fases. À medida que as tarefas vão sendo criadas, elas vão se encaixando em cada fase do projeto. Esse tipo de tratamento é uma opção a mais para o gerente de projetos utilizar no decorrer do projeto.

Sua usabilidade mostra ser bastante eficiente, tornando ao usuário um fácil manuseio do ambiente. Uma evidência disso é que ele centraliza todas as informações do projeto em uma página, mostrando reuniões, debates, anotações, arquivos, entre outros.

O ambiente poderia ter mais conteúdo em algumas de suas funcionalidades, como por exemplo, o lançamento de horas nas atividades dos projetos. Essa opção é um meio de extrema importância para o gerente de projetos, pois ele pode acompanhar o andamento das atividades da equipe do projeto em tempo real.

#### 4.4 Comparação dos Ambientes

Baseado na análise dos ambientes acima foram listadas as funcionalidades que cada ambiente suporta ou não.

Tabela 2 Funcionalidades de cada ambiente

Funcionalidade	DotProject	ClockingIt	NetOffice
Projetos	X	X	X
Tarefas	X	X	X
Lançamento de Horas	X	X	

<b>Eventos</b>	X		X
<b>Cadastro de Empresas</b>	X	X	X
<b>Calendário</b>	X	X	X
<b>Contatos</b>	X		
<b>Arquivos</b>	X	X	X
<b>Wiki / Bate Papo</b>		X	
<b>Multiusuário</b>	X	X	X
<b>Permissões</b>	X	X	X
<b>Fórum de Discussão</b>	X	X	
<b>Suporte Cliente</b>	X		
<b>Relatórios</b>	X	X	X

A Tabela 4 mostra as funcionalidades suportadas por cada ambiente analisado, considerando o marcador 'X' como funcionalidade presente no ambiente e considerando sem marcador para funcionalidade não presente no ambiente.

Como observado, o ambiente DotProject é o que mais suporta funcionalidades entre os ambientes observados. Tal fato não implica que ele seja o melhor ambiente para ser utilizado, pois isso depende muito do tipo de utilização e do objetivo desejado.

Os ambientes ClockingIt e NetOffice se mostraram mais simples quanto ao conteúdo de suas funcionalidades. Contudo, a usabilidade e interface desses ambientes se mostram em um nível bem mais avançado que do ambiente DotProject.

## 5 DESENVOLVIMENTO DO MODELO DO BANCO DE DADOS PARA O NOVO AMBIENTE DE GERÊNCIA DE PROJETOS

Agora é proposto um modelo de banco de dados relacional para ser utilizada como base de um novo ambiente *online* de gerência de projetos. O novo modelo é proposto baseado na observação dos ambientes analisados, conforme a seção anterior.

Inicialmente foram levantadas as necessidades que o sistema deveria atender. Foram definidas então as funcionalidades do novo modelo de banco de dados. Para facilitar o entendimento, as funcionalidades foram divididas em dois tipos de usuários: usuário comum e o usuário administrador.



Figura 33 Funcionalidades disponíveis para o usuário comum

A Figura 33 demonstra as funcionalidades para o usuário comum, ou seja, aquele tipo de usuário que realiza operações oferecidas pelo ambiente de gerência de projetos.



Figura 34 Funcionalidades disponíveis para o usuário administrador

A Figura 34 demonstra as funcionalidades para o usuário administrador, ou seja, o usuário responsável pela manutenção e administração de permissões, usuários e configurações do ambiente.

Após o levantamento das funcionalidades, foi criado um modelo da nova estrutura de banco de dados. No total, foram criadas 29 tabelas, que serão mostradas nas seções seguintes.

## 5.1 Modelagem do Banco de Dados

Segue agora uma descrição completa do funcionamento do novo modelo de banco de dados desenvolvido. A representação escolhida para demonstrar a estrutura foi a UML. O modelo desenvolvido foi dividido em

algumas funcionalidades para facilitar a explicação. Para consultar o modelo completo, consultar o Apêndice B.

Para cada funcionalidade serão mostrados dois tipos de visualização gráfica. A primeira consiste nas tabelas de banco de dados como sendo objetos com o objetivo de contextualizar a situação, ou seja, representando o modelo conceitual. A segunda visualização consiste em uma estrutura UML das tabelas criadas e seus relacionamentos, representando o modelo de implementação. Para informações ainda mais detalhadas de cada campo das tabelas é recomendada a consulta ao Apêndice A.

Antes, seguem algumas definições de notação do modelo UML utilizado para demonstrar as estruturas do banco de dados.

A definição dos campos em uma tabela do modelo mantém um padrão de classificação específico.

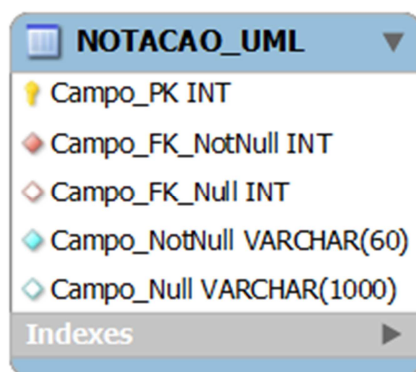


Figura 35 Exemplo de estrutura de uma tabela no modelo UML

A Figura 35 representa um exemplo de tabela do modelo UML. Neste caso, denominada de NOTACAO\_UML, possui cinco campos. A notação de um campo de uma tabela no modelo é representado por três partes: um ícone, um nome e o tipo do campo. O ícone é utilizado para caracterizar o tipo de relação (chave primária ou estrangeira) e a obrigatoriedade de preenchimento ou

não (nulo e não nulo). O nome para nomenclatura do campo. O tipo define o conteúdo do campo, como por exemplo, string, numérico, data, bit, entre outros.

No exemplo da Figura 35 são demonstrados todos os tipos de utilização de um campo, no que se diz respeito aos tipos de ícones. O primeiro campo, com um ícone de uma chave dourada antes do nome, representa que aquele campo é uma chave primária da tabela. O segundo campo, representado por um losango da cor vermelha, significa que o campo possui uma chave estrangeira e é considerado não-nulo (Not Null). O terceiro campo, representado por um losango da cor branca e com borda vermelha, significa que o campo possui uma chave estrangeira e é considerado nulo (Null). O quarto campo, representado por um losango da cor azul, significa que o campo é considerado não-nulo (Not Null). O quinto campo, representado por um losango da cor branca e com borda azul, significa que o campo é considerado nulo (Null).

Quanto aos relacionamentos entre tabelas, utilizados no modelo de implementação, são apresentados dois tipos de relacionamentos.

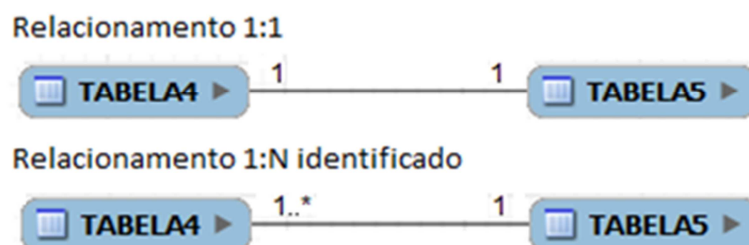


Figura 36 Tipos de relacionamentos entre tabelas

O exemplo da Figura 36 demonstra os tipos de relacionamentos utilizados no modelo de banco de dados.

- **Relacionamento 1:1** - significa que um registro da TABELA4 está associado à no máximo um registro da TABELA5 e vice-

versa. Além disso, o campo da tabela relacionada pode ser uma chave primária.

- **Relacionamento 1:N** - significa que um registro da TABELA5 está associado a qualquer número de registros da TABELA4, entretanto um registro da TABELA4 está associado a no máximo um registro da TABELA5. Além disso, o campo da tabela relacionada pode ser uma chave primária.

### 5.1.1 Multiusuário / Permissões

Iniciando a descrição das funcionalidades do sistema, tem-se a funcionalidade de multiusuários e permissões do ambiente. De uma maneira geral, ela permite a criação de vários usuários com os mais variados níveis de permissões.



Figura 37 Relações entre tabelas de usuário

A Figura 37 mostra as relações entre os objetos da funcionalidade Multiusuário / Permissões. De um modo geral, todo usuário tem uma função ou cargo e pertence a um setor ou módulo. Já um setor ou módulo deve ter um usuário como responsável.

As permissões foram classificadas em dois tipos: as permissões de ambiente e as permissões de projetos. Nessa funcionalidade, será tratada apenas a permissão de ambiente.

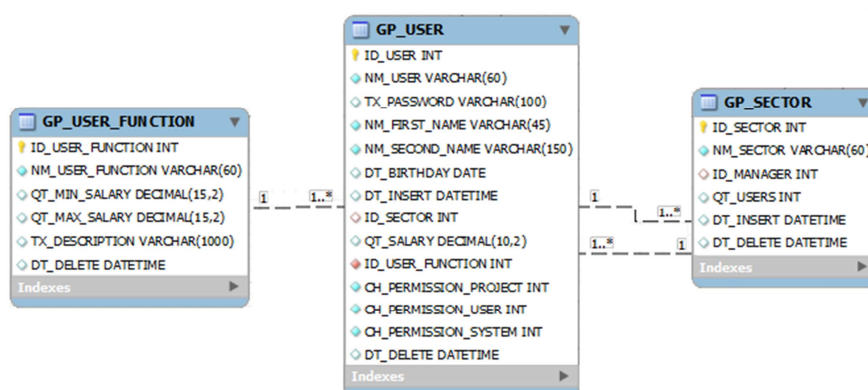


Figura 38 Tabelas para Usuários / Permissões

A Figura 38 mostra as tabelas e os campos utilizados para cadastro de usuários e permissões. É permitido então adicionar usuários no sistema e para isso foi criada a tabela GP\_USER, sendo possível detalhar nome, uma senha qualquer, data de nascimento, data de inserção e deleção do registro do usuário no sistema, salário, setor da empresa e função do usuário.

Para o cadastro de função do usuário, foi criada a tabela GP\_USER\_FUNCTION, a qual irá armazenar todos os possíveis cargos da empresa, com nome, faixa salarial e descrição. Quando um usuário for inserido, então é buscado um cargo nessa tabela para ele.

Existe também a possibilidade de cadastrar setores da empresa, na tabela GP\_SECTOR. A ideia é cadastrar todos os setores da empresa com nome, responsável pelo setor, quantidade de usuários e data de inserção e deleção do registro na tabela. Quando um usuário for cadastro, então é buscado o seu setor nessa tabela.

É possível identificar também três campos para o tratamento de permissão de usuário no ambiente, são eles: CH\_PERMISSION\_PROJECT, CH\_PERMISSION\_USER e CH\_PERMISSION\_SYSTEM. A ideia é preencher nesses campos os valores '1' ou '0' para cada usuário, sendo que ao



ter o valor '1' cadastrado, o usuário tem permissão para realizar tal competência e o valor '0' o usuário não tem permissão. Os significados desses campos são:

- CH\_PERMISSION\_PROJECT: Indica se o usuário tem permissão para criar projetos dentro do ambiente.
- CH\_PERMISSION\_USER: Indica se o usuário tem permissão para adicionar outros usuários no ambiente.
- CH\_PERMISSION\_SYSTEM: Indica se o usuário tem permissão para ajustar as configurações do sistema.

### 5.1.2 Cadastro de Empresas

A funcionalidade de cadastro de empresas / clientes permite o cadastro de organizações ligadas ao cotidiano do usuário.

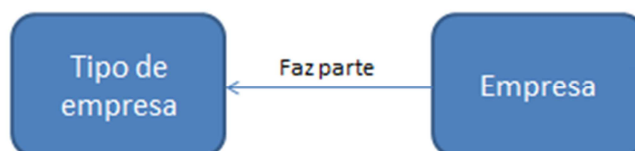


Figura 39 Relações entre tabelas de empresa

É permitido detalhar várias características na hora de armazenar determinada organização. A Figura 39 mostra que toda empresa faz parte de um tipo de empresa (cliente, fornecedor, parceiros, entre outros).

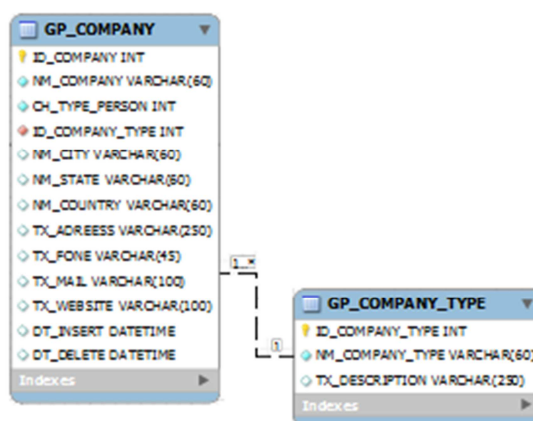


Figura 40 Tabelas para Cadastro de Empresas

A Figura 40 mostra as duas tabelas utilizadas para tratar a funcionalidade de cadastro de empresas. Foi criada a tabela GP\_COMPANY para dar suporte a gravação de empresas, onde é possível cadastrar seu código identificador, nome, tipo de pessoa (física ou jurídica), tipo da empresa, endereço de origem, telefone, email, website e data de inserção e deleção do registro no banco de dados.

Para realizar o tratamento do tipo de empresa, foi criada uma tabela, GP\_COMPANY\_TYPE, que irá armazenar os tipos possíveis de empresas a serem cadastradas. A ideia dessa tabela é poder definir diferentes tipos de utilização das empresas, não tratando somente clientes, mas também fornecedores e outros. Por default, foram criados cinco tipos de empresas, conforme a Tabela 3.

Tabela 3 Registros default de tipos de empresas

ID	Tipo de Empresa	Descrição
1	Empresa Pública	Empresas do setor público
2	Empresa Privada	Empresas do setor privado

3	Fornecedor	Empresa fornecedora de recursos
4	Governo	Setores do governo federal
5	Particular	Cliente particular

Esses registros default podem ser modificados conforme a necessidade do usuário. Existe também a possibilidade de adicionar outros registros para tratar outros tipos de empresas.

### 5.1.3 Projetos

A funcionalidade de cadastro de projetos permite a manipulação de projetos, seja adição, modificação e exclusão. Na criação de um projeto, é possível detalhar várias características, conforme a estrutura da Figura 37.

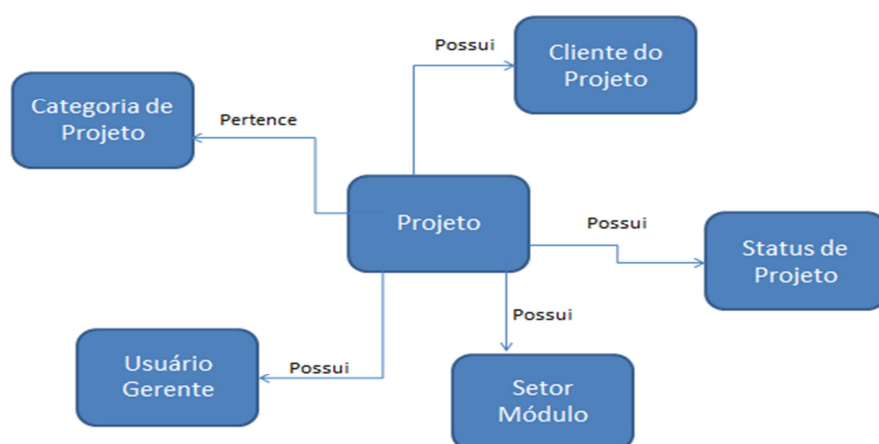


Figura 41 Relações entre as tabelas de Cadastro de Projetos

A Figura 41 mostra as relações entre as tabelas utilizadas pela funcionalidade. Ao criar um projeto, deve-se especificar um cliente, um setor da

empresa responsável pelo projeto, um gerente de projetos, uma categoria e por fim, seu status.

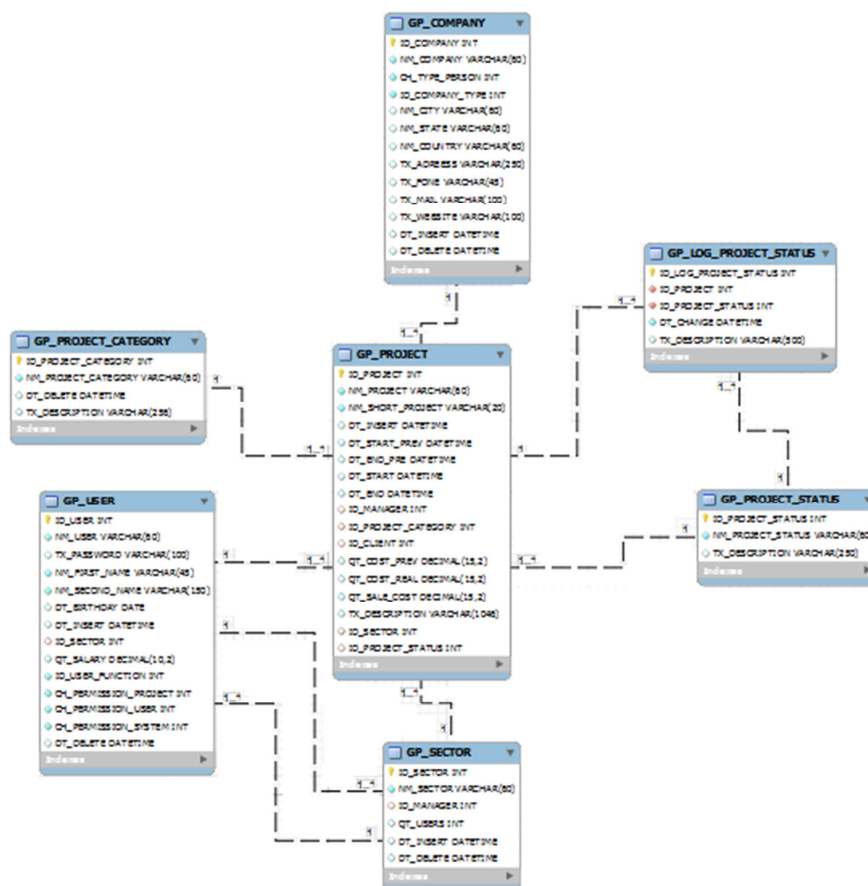


Figura 42 Tabelas para Cadastro de Projetos

A Figura 42 mostra a estrutura das tabelas para se criar um projeto. Os projetos serão armazenados na tabela GP\_PROJECT, onde é permitido armazenar seu código identificador, nome, abreviatura, data de inserção do registro, data de início / fim previsto e realizado, gerente do projeto, cliente, custo previsto e real, preço de venda, setor interno da empresa responsável pelo desenvolvimento do projeto, status e descrição.

Para armazenar o gerente responsável pelo projeto, foi criada uma relação com a tabela GP\_USER. Para guardar o setor interno da empresa responsável pelo desenvolvimento do projeto, foi criada uma relação com a tabela GP\_SECTOR. Para armazenar o cliente do projeto, foi criada uma relação com a tabela GP\_COMPANY. Todas essas tabelas já foram descritas anteriormente.

A tabela GP\_PROJECT\_CATEGORY foi criada para permitir o cadastro de categorias de projetos, sendo possível cadastrar seu código identificador, nome da categoria de projeto, descrição e data de deleção do registro do banco de dados. Por default, foram criadas cinco categorias de projetos, conforme a Tabela 4.

Tabela 4 Registros default de categorias de projeto

ID	Categoria de Projeto	Descrição
1	Projeto de Implantação	Tipo de projeto para implantação de um produto pronto no cliente.
2	Projeto Customizado	Tipo de projeto para realizar customização no produto do cliente.
3	Projeto de Novo Produto	Tipo de projeto para desenvolver um novo produto.
4	Projeto Interno	Tipo de projeto interno a organização.
5	Consultoria	Tipo de projeto para serviços de consultoria.

Esses registros default podem ser modificados conforme a necessidade do usuário. Existe também a possibilidade de adicionar outros registros para tratar outras categorias de projeto.

A tabela GP\_PROJECT\_STATUS foi criada para permitir o cadastro de status dos projetos, sendo possível cadastrar seu código identificador, nome do status do projeto e sua descrição. Por default, foi criado cinco status de projetos, conforme a Tabela 5.

Tabela 5 Registros default de status de projeto

ID	Status do Projeto	Descrição
1	Aberto	Projeto criado recentemente
2	Em Desenvolvimento	Projeto em desenvolvimento
3	Finalizado	Projeto finalizado
4	Suspenso	Projeto suspenso
5	Em negociação	Projeto em negociação com o cliente

Os registros default da Tabela 5 podem ser modificados conforme a necessidade do usuário. Existe também a possibilidade de adicionar outros registros para tratar outros status de projeto.

A tabela GP\_LOG\_PROJECT\_STATUS foi criada com o objetivo de armazenar o log de alteração de status dos projetos. Sua estrutura se dá pelo campo identificador do log, pelo código do projeto, o código do status do projeto, a data de alteração do status do projeto e uma descrição do motivo de troca de status.

Outros pontos importantes da funcionalidade de projetos são as metas de projetos, os custos de projeto e os membros com suas permissões de projetos.

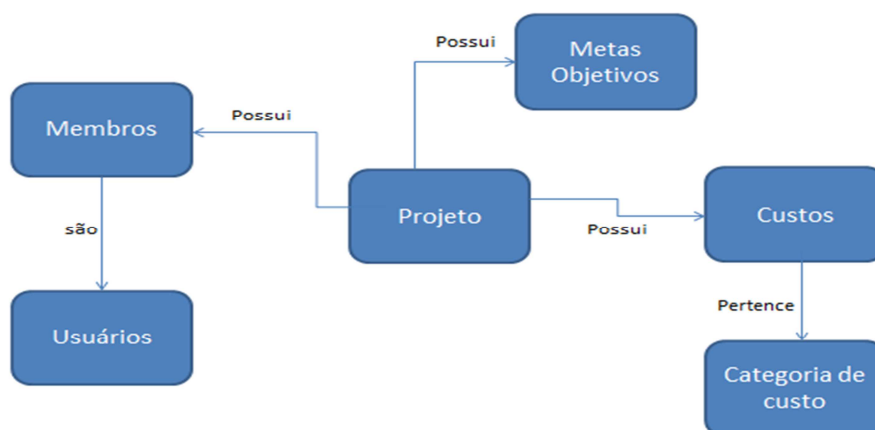


Figura 43 Relações entre as tabelas de Projetos

A Figura 43 representa as relações entre metas, membros e custos de projetos. A Figura 44 mostra a estrutura das tabelas criadas para tornar possível essas operações.

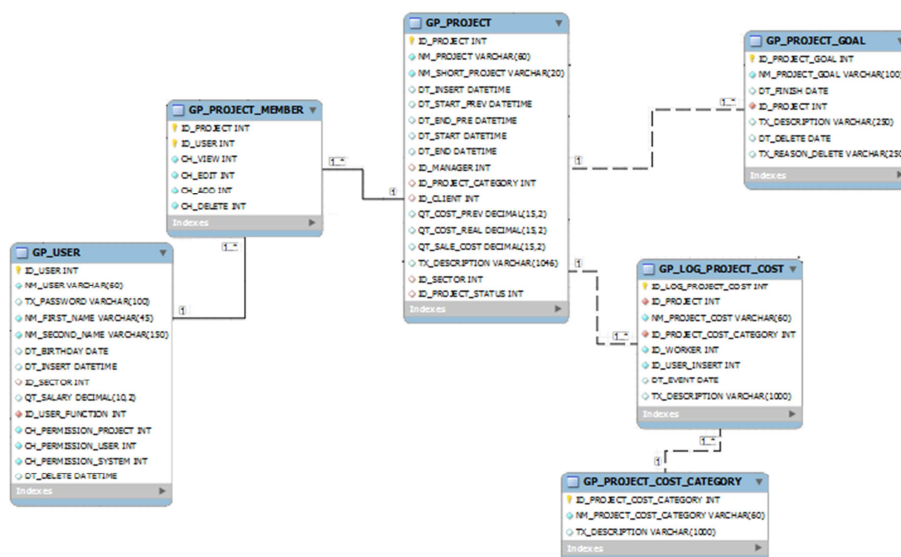


Figura 44 Outras tabelas da funcionalidade de Projetos

Tomando como base as metas do projeto, a ideia é permitir ao usuário cadastrar metas de projetos para que toda a equipe tenha conhecimento e se empenhem em atingí-las. A tabela criada foi a GP\_PROJECT\_GOAL, onde é possível detalhar o código identificador da meta, nome, data prevista para alcance da meta, projeto ao qual ela pertence (relação com a tabela GP\_PROJECT), descrição, data de deleção da meta e a descrição do motivo de cancelamento da meta, caso houver.

Outro ponto importante e tratado na estrutura é o de custos de projeto. O objetivo desse tratamento é dar ao usuário um maior controle dos custos, tendo um histórico de todos os lançamentos de custos realizados no decorrer do projeto. Além disso, os custos serão tratados por categorias. As tabelas criadas

para dar suporte a esse funcionamento foram a GP\_LOG\_PROJECT\_COST e a GP\_PROJECT\_COST\_CATEGORY.

A tabela GP\_LOG\_PROJECT\_COST é uma tabela de histórico de lançamentos de custos. Toda vez que um membro da equipe realizar um custo, então este deve ser lançado e caracterizado. A tabela armazena então o código identificador do custo, o código do projeto ao qual o custo pertence, nome, categoria do custo, usuário que realizou o custo, usuário que cadastrou o custo no sistema, data do custo e descrição.

Para armazenar as categorias de custos de projetos, foi criada a tabela GP\_PROJECT\_COST\_CATEGORY, que permite o armazenamento do seu código identificador, nome da categoria e descrição. Por default, foram criados seis exemplos de categorias de custo de projeto, conforme a Tabela 6.

Tabela 6 Registros default de categoria de custo de projeto

ID	Categoria de Custo	Descrição
1	Deslocamento	Custos envolvendo deslocamentos de um local para o outro
2	Alimentação	Custos envolvendo alimentação
3	Compra Hardware	Custos envolvendo compra de hardware
4	Compra Software	Custos envolvendo compra de software
5	Horas Extras	Custos envolvendo horas extras da equipe
6	Comunicação	Custos envolvendo comunicação com os envolvidos no projeto

Os registros default da Tabela 6 podem ser modificados conforme a necessidade do usuário. Existe também a possibilidade de adicionar outros registros para tratar outras categorias de custo de projeto.

Um último ponto a ser considerado na funcionalidade projetos é a associação de usuários nos projetos e suas permissões. Foi criada a tabela



GP\_PROJECT\_MEMBER, que é uma tabela de associação, onde cada registro corresponde a uma associação usuário / projeto e suas permissões. É possível armazenar na tabela de associação o código do usuário, código do projeto ao qual o usuário está participando, permissão de ver o projeto, permissão de editar o projeto, permissão de adicionar itens ao projeto e a permissão de apagar itens do projeto. Esse tipo de tabela associativa contém uma chave primária dupla, ou seja, não é possível inserir um registro onde os valores dos dois campos definidos como chave sejam iguais.

Os outros quatro campos existentes na tabela GP\_PROJECT\_MEMBER configuram o segundo tipo de permissão do ambiente, onde é possível configurar a permissão dos usuários dentro dos projetos. Para cada registro da tabela, que corresponde à participação de um usuário em determinado projeto, deverá ser preenchido '0' ou '1' nos campos de permissão, quando o registro possuir o valor '1', significa que o usuário terá a permissão de fazer determinada operação dentro do projeto, caso contrário, não terá a permissão. Abaixo, segue a definição de cada campo de permissão de projeto.

- CH\_VIEW: Indica se o usuário terá permissão para visualizar o projeto (1) ou não (0).
- CH\_EDIT: Indica se o usuário terá permissão para editar os itens do projeto (1) ou não (0).
- CH\_ADD: Indica se o usuário terá permissão para adicionar os itens do projeto (1) ou não (0).
- CH\_DELETE: Indica se o usuário terá permissão para apagar os itens do projeto (1) ou não (0).

É importante lembrar que essas permissões ficam restritas apenas a um determinado projeto, ou seja, um usuário pode ter um tipo de permissão em determinado projeto e outros tipos de permissão em outros projetos.

### 5.1.4 Tarefas

A funcionalidade de cadastro de tarefas dá suporte ao cadastro de tarefas dos projetos criados. As tarefas permitem uma maior organização e sequência do projeto.

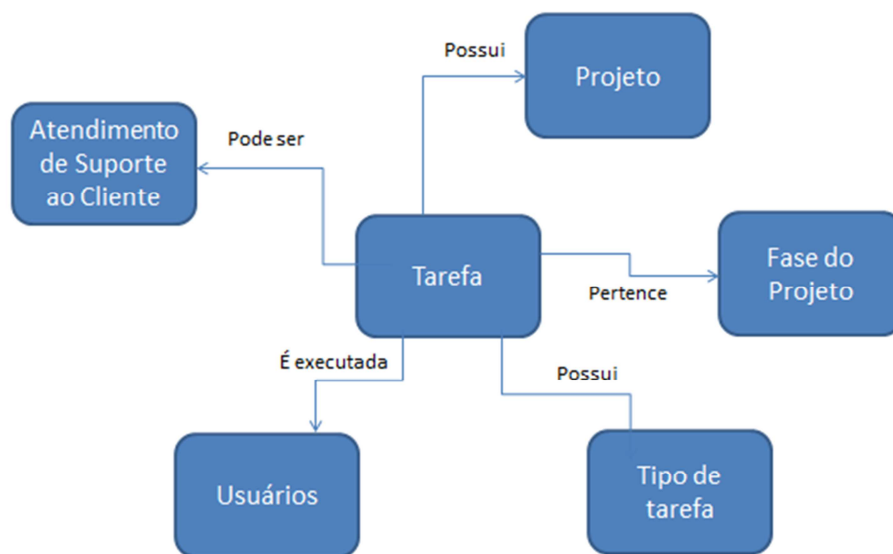


Figura 45 Relações entre tabelas da funcionalidade de tarefas

A Figura 45 mostra que ao cadastrar uma tarefa ela deve possuir um projeto, pertencer a uma fase do projeto e possuir um tipo de tarefa. Além disso, ela pode ser delegada a um determinado usuário ou ser uma tarefa de atendimento de suporte ao cliente.

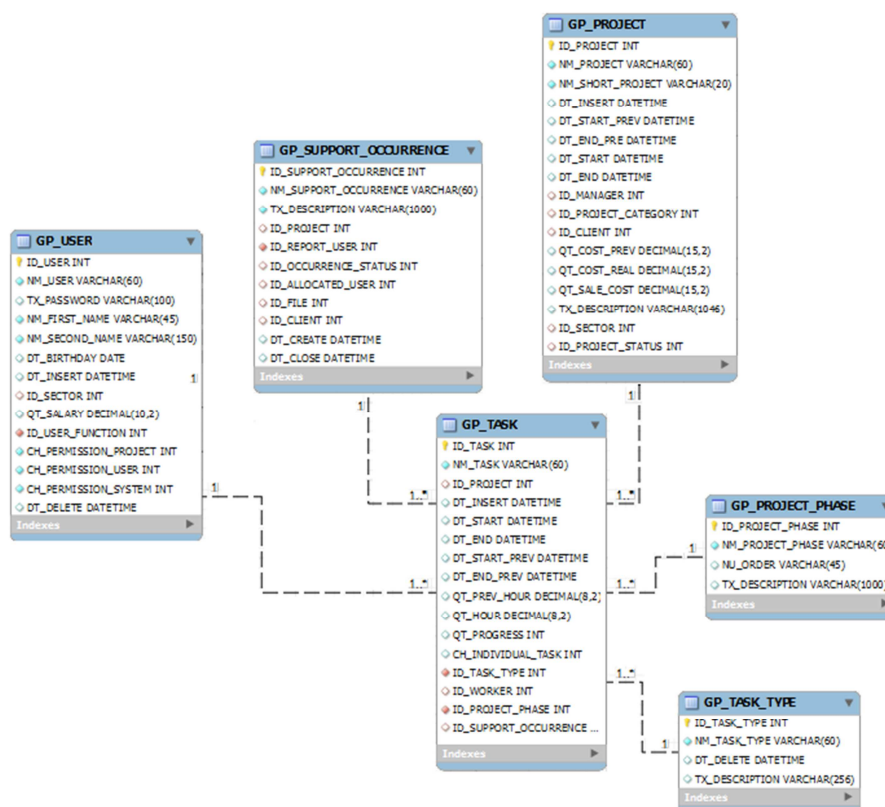


Figura 46 Tabelas da funcionalidade de Cadastro de Tarefas

A Figura 46 mostra a estrutura criada para o cadastro de tarefas de projetos. A tabela criada para armazenar as tarefas foi a GP\_TASK, que permite armazenar as características de cada tarefa. É possível caracterizar seu código identificador, nome da tarefa, projeto ao qual a tarefa pertence, data de criação, data de início/fim prevista e real, horas previstas e gastas, identificador de tarefa individual ou coletiva, código do usuário alocado para a tarefa, fase do projeto a qual a tarefa pertence e código da ocorrência de suporte, caso a tarefa seja de atendimento ao cliente.

Para armazenar o usuário alocado para a tarefa, foi criada uma relação com a tabela GP\_USER. Para armazenar o projeto ao qual a tarefa pertence, foi criada uma relação com a tabela GP\_PROJECT. Para guardar o código de atendimento ao cliente ao qual a tarefa está relacionada, foi criada uma relação com a tabela GP\_SUPPORT\_OCCURRENCE.

As fases de projeto foi um conceito utilizado para tratar tarefas e projetos na estrutura do ambiente. A ideia é possibilitar uma melhor referência ao envolvidos no projeto sobre o que cada tarefa tem a ver com cada fase do projeto. Para isso, foi criada a tabela GP\_PROJECT\_PHASE, tabela que irá armazenar as fases de um projeto. É permitido armazenar o seu código identificador, nome da fase, sua ordem numérica e sua descrição. Por default, foram criadas sete fases de um projeto, conforme a Tabela 7.

Tabela 7 Registros default de fases de projeto

ID	Nome da fase	Descrição
1	Concepção	Fase inicial do projeto. Definição da missão e objetivo do projeto.
2	Planejamento	Fase de planejamento do projeto. Definição das estratégias, cronogramas, alocações, entre outros.
3	Desenvolvimento	Fase de desenvolvimento do que foi planejado.
4	Monitoramento e Controle	Fase que pode ocorrer paralelamente ao desenvolvimento. Tem como objetivo acompanhar e controlar o que está sendo realizado.
5	Teste	Fase de testes integrados do que foi desenvolvido antes de ser implantado.

6	Documentação	Fase para documentação do que foi desenvolvido. Pode ser incluído nessa fase processos de rollout, merge, entre outros.
7	Manutenção	Fase de manutenção de possíveis problemas após a implantação do projeto no cliente.

Os registros default da Tabela 7 podem ser modificados conforme a necessidade do usuário. Existe também a possibilidade de adicionar outros registros para tratar outras fases de projeto.

### 5.1.5 Lançamento de Horas

A funcionalidade de lançamento de horas permite ao usuário detalhar todo o tempo gasto nas atividades e tarefas dos projetos. A ideia é armazenar todos os lançamentos de horas de cada usuário como um registro em uma tabela de log. Esse log é semelhante a um histórico, permitindo ao usuário acompanhar o andamento de cada tarefa desenvolvida e tomar decisões apropriadas quando necessário.

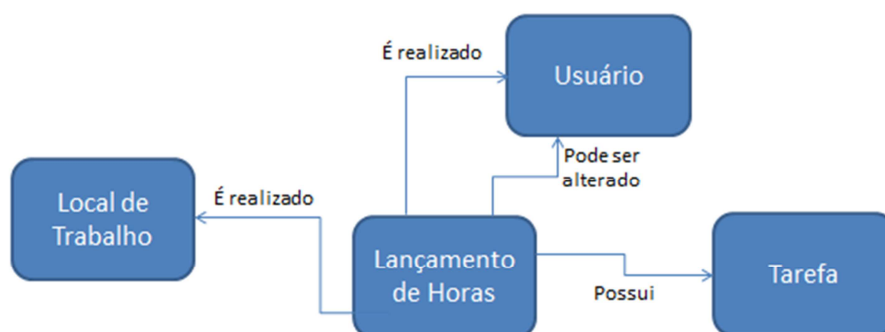


Figura 47 Relações entre tabelas da funcionalidade de lançamento de horas

A Figura 47 mostra que um lançamento de horas é o registro de atividade do usuário dentro de uma tarefa do projeto. O lançamento de horas deve ser executado por um usuário, que deve especificar o local de trabalho onde foi realizada a tarefa. Caso necessário, um usuário pode alterar algumas características da atividade realizada.

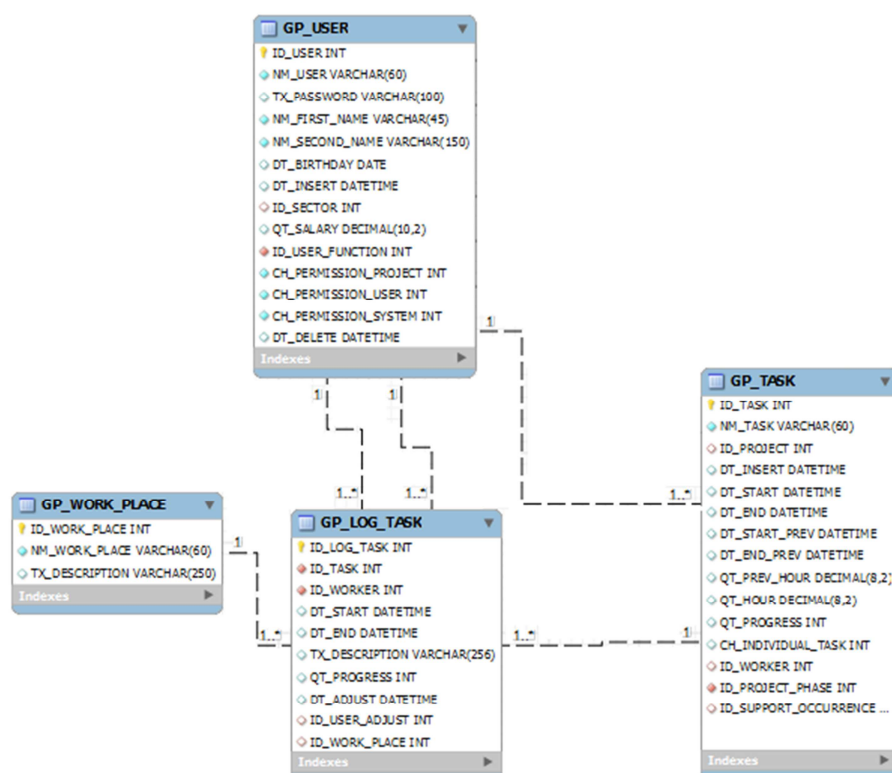


Figura 48 Tabelas da funcionalidade de lançamento de horas

Conforme a Figura 48, a tabela criada para armazenar o log de lançamento de horas é a GP\_LOG\_TASK, onde é possível armazenar o código do log, o código da tarefa de projeto, o código do usuário que realizou o lançamento, data de início/fim, descrição do que foi feito durante a atividade,

progresso atualizado da tarefa, código de onde foi realizado o trabalho e, por fim, caso o lançamento precise de ajuste, tem-se a data de ajuste e o usuário que realizou o ajuste do lançamento.

Para armazenar o usuário que realizou o lançamento de horas e o usuário que realizou o ajuste do lançamento, foi criada duas relações para a tabela GP\_USER. Já para armazenar a tarefa ao qual o lançamento pertence, foi criada a relação com a tabela GP\_TASK.

Como citado anteriormente, a tabela de log também armazena o código de onde o trabalho foi realizado. Para isso, foi criada a tabela GP\_WORK\_PLACE, que armazena os lugares possíveis de trabalho do funcionário. Além do seu código identificador, é possível armazenar o nome e a descrição do local. Por default, foi criado quatro exemplos de locais de trabalho, conforme a Tabela 8.

Tabela 8 Registros default de locais de trabalho

ID	Local de Trabalho	Descrição
1	Empresa	Trabalho realizado dentro da sede da empresa
2	In Loco	Trabalho realizado nas dependências do cliente
3	Deslocamento	Trabalho realizado em período de viagens
4	Trabalho em casa	Trabalho realizado na casa do funcionário

Os registros default da Tabela 8 podem ser modificados conforme a necessidade do usuário. Existe também a possibilidade de adicionar outros registros para tratar outros locais de trabalho.

### 5.1.6 Eventos / Calendário

A funcionalidade de eventos / calendário foi desenvolvida com o objetivo de proporcionar ao usuário uma maior facilidade na organização de sua rotina no que se diz respeito a eventos e compromissos. A parte que requer um tratamento especial na estrutura de banco de dados é quanto ao cadastro de eventos, onde a existência de tabelas específicas é necessária para dar suporte ao armazenamento dos registros cadastrados. A parte de calendário consiste em realizar uma busca nas tabelas de eventos para verificar os compromissos de cada usuário. Outro ponto que pode ser considerado em calendários, é a busca das alocações de projetos de cada usuário.

A estrutura desenvolvida permite o cadastro de eventos, podendo relacionar o mesmo a projetos específicos ou não.

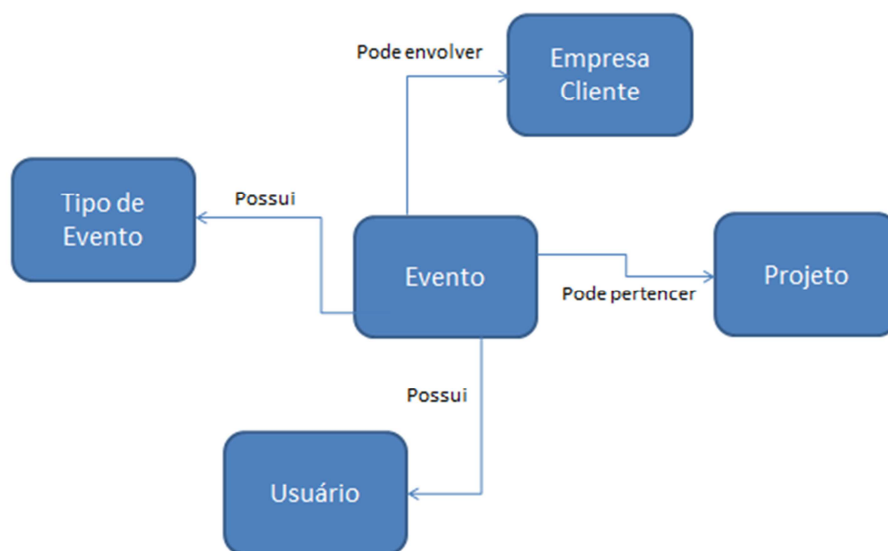


Figura 49 Funcionamento da funcionalidade de eventos



A Figura 49 mostra o funcionamento do cadastro de eventos. Ao criar um evento, é possível convidar usuários, relacionar a projetos e empresas; e definir seu tipo.

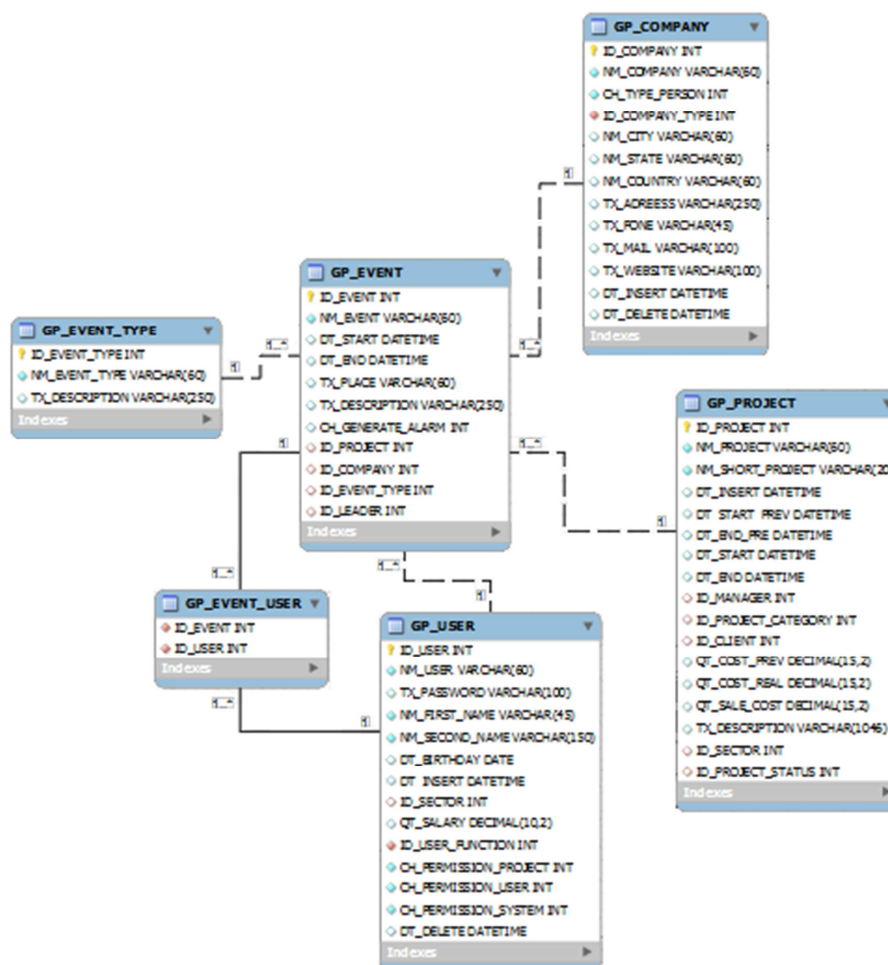


Figura 50 Tabelas da funcionalidade de Eventos

A Figura 50 mostra a estrutura desenvolvida para a funcionalidade de Eventos. A tabela utilizada para cadastrar eventos foi GP\_EVENT. Ela permite a caracterização de diversos detalhes do evento, tais como seu código

identificador, nome, data de início/fim, local, descrição, opção para gerar alarme, código do projeto, código do cliente, tipo de evento e o código do usuário responsável pelo evento.

Para armazenar o código do projeto, criou-se uma relação com a tabela GP\_PROJECT. Para armazenar o código do cliente, criou-se uma relação com a tabela GP\_COMPANY. Para armazenar o código do usuário responsável pelo evento, criou-se uma relação com a tabela GP\_USER.

Nessa funcionalidade, os eventos serão classificados por tipo de evento. Para isso, foi criada a tabela GP\_EVENT\_TYPE que armazenará o código identificador do tipo de evento, o nome do tipo de evento e a sua descrição. Por default, foram criados quatro exemplos, conforme a Tabela 9.

Tabela 9 Registros default de tipos de eventos

<b>ID</b>	<b>Tipo de Evento</b>	<b>Descrição</b>
<b>1</b>	Reunião Interna	Reunião de módulos internos a empresa
<b>2</b>	Reunião com Cliente	Reunião com clientes da empresa
<b>3</b>	Reunião de Projeto	Reunião de determinado projeto
<b>4</b>	Reunião com Fornecedor	Reunião com fornecedores da empresa

Os registros default da Tabela 9 podem ser modificados conforme a necessidade do usuário. Existe também a possibilidade de adicionar outros registros para tratar outros tipos de eventos.

Por fim, a última tabela necessária para o funcionamento da funcionalidade de eventos é a GP\_EVENT\_USER, que é uma tabela de associação de eventos e usuários, ou seja, para cada registro tem-se um evento associado a um usuário. Essa tabela possui uma chave primária dupla, não permitindo assim combinações de usuários e eventos idênticos. Com uma consulta nessa tabela, é possível identificar todos os usuários participantes de um determinado evento.

### 5.1.7 Contatos

A funcionalidade de contatos foi criada com a ideia de cada usuário ter sua lista de contatos. A figura 51 tem a estrutura para o funcionamento da funcionalidade.

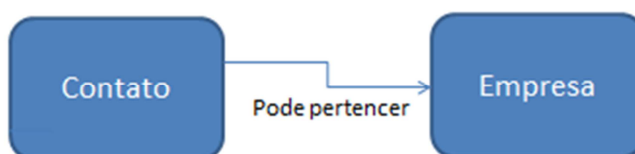


Figura 51 Relações da funcionalidade de Contatos

A Figura 51 representa que ao criar um contato, ele pode pertencer a uma empresa cadastrada no ambiente.

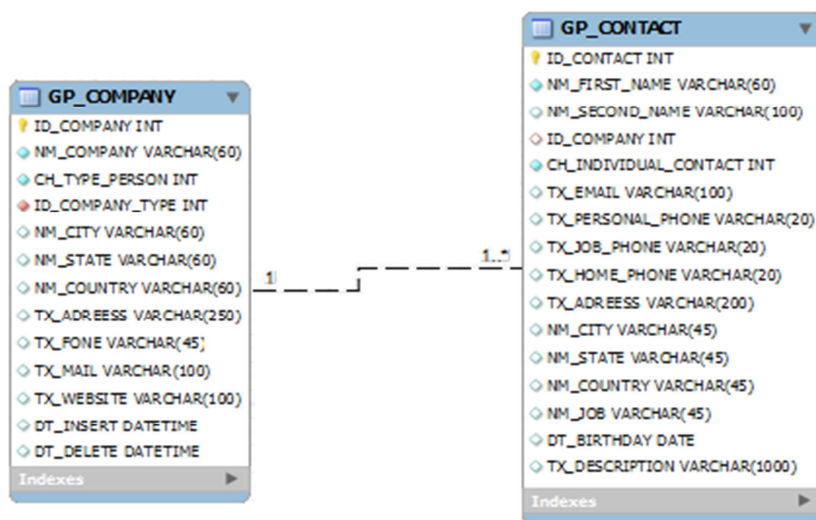


Figura 52 Tabelas da funcionalidade de Contatos

Como mostra a Figura 52, foi criada a tabela GP\_CONTACT para armazenar os contatos cadastrados pelo usuário. É possível armazenar seu código identificador, nome do contato, sobrenome, código da organização cadastrada no sistema (caso exista), campo identificador de contato individual ou público, e-mail, telefones para contato, endereço, cidade, estado, país, cargo de trabalho, data de aniversário e descrição do contato.

Para armazenar o código da organização, foi criada uma relação com a tabela GP\_COMPANY.

### 5.1.8 Arquivos

A funcionalidade de arquivos foi criada para o usuário poder adicionar arquivos relacionados a projetos e ao cotidiano da organização. A estrutura de banco de dados criada para arquivos dará suporte à linguagem de programação no sentido de detalhar o arquivo armazenado, ou seja, assim que um arquivo for armazenado no ambiente, então a tabela GP\_FILE é preenchida com as características do mesmo.

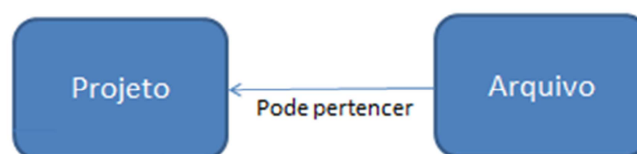


Figura 53 Relação da funcionalidade de Arquivos

A Figura 53 mostra que um arquivo pode pertencer a um projeto. O objetivo é que todos os membros do projeto possam ter acesso a esse arquivo.

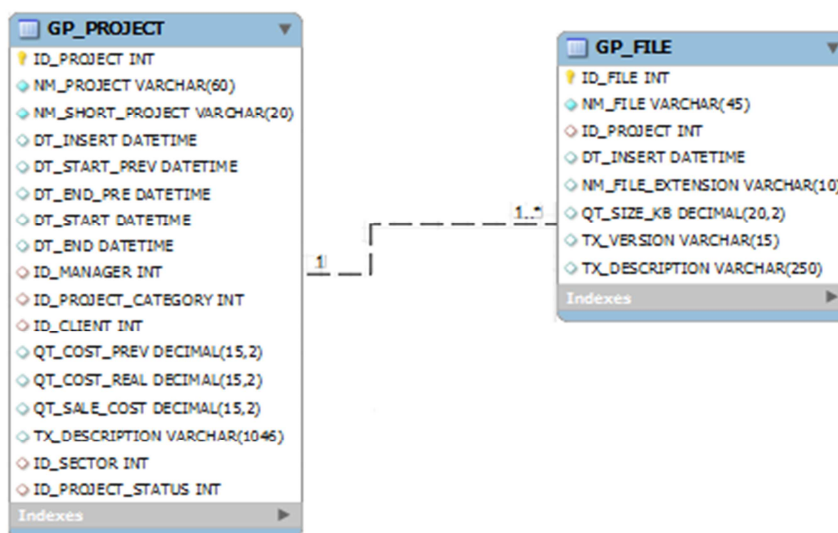


Figura 54 Tabelas da funcionalidade de Arquivos

A Figura 54 mostra a tabela GP\_FILE criada para armazenar as características e informações dos arquivos colocados no ambiente. É possível detalhar seu código identificador, nome do arquivo, código do projeto (caso houver), data de inserção do arquivo, extensão do arquivo, tamanho, versão e descrição.

Para armazenar o código do projeto, foi criada uma relação com a tabela GP\_PROJECT. A ideia de associar o arquivo a um projeto é no sentido de permitir o acesso somente a membros do projeto.

### 5.1.9 Fórum de Discussão

A funcionalidade de fórum de discussão foi criada para facilitar a comunicação entre os membros dos projetos e de toda a organização. A ideia é criar tópicos reservados ou não para interação dos usuários.

A estrutura de banco de dados montada para essa funcionalidade partiu de três conceitos: fóruns, tópicos e mensagens. O primeiro passo é a criação de um fórum, como por exemplo, fórum de dúvidas. Após a criação do fórum, é necessário a criação de um tópico do fórum, como por exemplo, como criar usuário. Com a criação do tópico, os usuários passariam a se comunicar através dos tópicos específicos de cada fórum. Dentro de um fórum existe a possibilidade de haver vários tópicos. É permitido também criar outros fóruns, como por exemplo, fórum de projetos, avisos, entre outros.

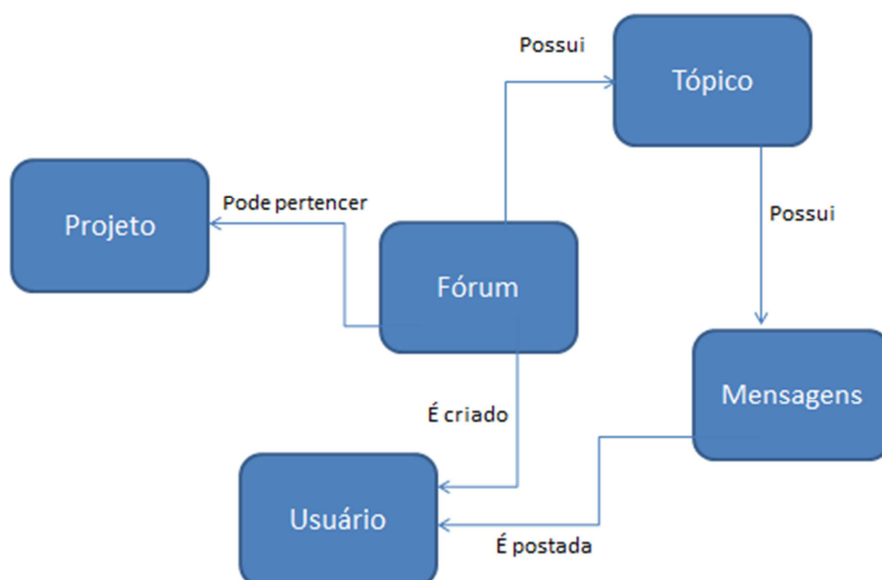


Figura 55 Representação da funcionalidade Fórum de Discussão

A Figura 55 representa que quando um fórum é criado ele pode ter tópicos, e dentro dos tópicos, mensagens de usuários. O fórum também pode ser relacionado a um projeto e apenas seus membros poderiam visualizar seu conteúdo.

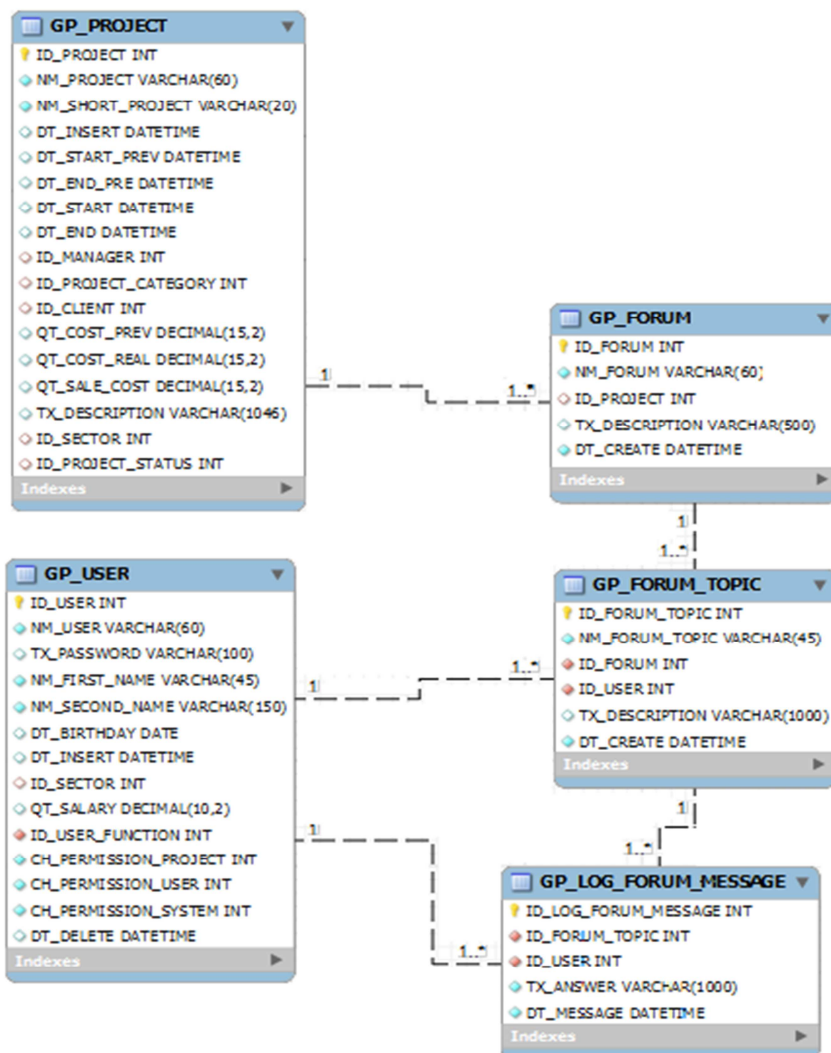


Figura 56 Tabelas da funcionalidade de Fórum de Discussão

Conforme a Figura 56, a tabela criada para armazenar os fóruns foi a GP\_FORUM, que permite detalhar o código identificador do fórum, o nome do fórum, o código do projeto (se for um fórum de projetos), descrição e a data de

criação. Para armazenar o código do projeto, foi criada uma relação com a tabela GP\_PROJECT.

Para armazenar os tópicos dos fóruns, foi criada a tabela GP\_FORUM\_TOPIC, que permite armazenar o código identificador do tópico, nome, código do fórum ao qual o tópico pertence, usuário que criou o tópico, descrição e data de criação. Para armazenar o código do fórum, foi criada uma relação com a tabela GP\_FORUM. Já para armazenar o código do usuário, foi criada uma relação com a tabela GP\_USER.

O armazenamento das mensagens será feita pela tabela GP\_LOG\_FORUM\_MESSAGE, que armazenará o código identificador da mensagem, o código do tópico de fórum, o usuário que criou a mensagem, o texto da mensagem e a data de postagem. Para armazenar o código do tópico de fórum, foi criada uma relação com a tabela GP\_FORUM\_TOPIC. Já para armazenar o código do usuário, foi criada uma relação com a tabela GP\_USER.

#### 5.1.10 Suporte ao Cliente

A funcionalidade de suporte ao cliente foi criada buscando agilizar o processo de atendimento deste.

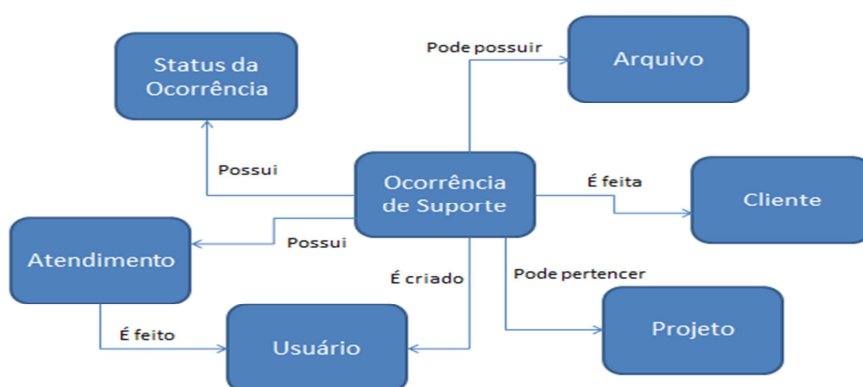


Figura 57 Relações da funcionalidade de suporte ao cliente



Como mostrado na figura 57, a estrutura montada permite ao cliente realizar o cadastro de chamadas de suporte, ou seja, criar uma espécie de tópico em uma determinada página solicitando um atendimento ou descrevendo um erro de produto ou projeto por exemplo. Após o cadastro, o usuário do sistema visualiza a solicitação do cliente e pode trabalhar no seu atendimento, sendo que ele também pode escrever uma mensagem e atualizar o status do atendimento para o cliente como forma de mantê-lo informado e atualizado.

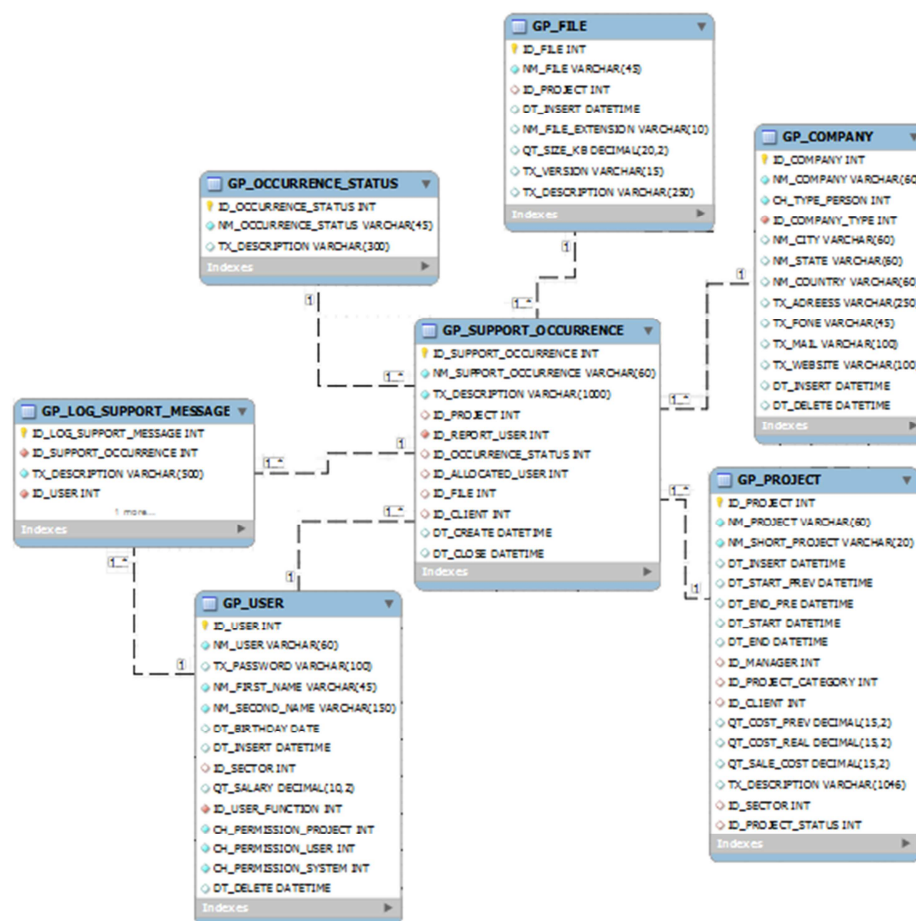


Figura 58 Tabelas da funcionalidade de Suporte ao Cliente

Conforme mostrado na Figura 58, foram criadas três tabelas para dar suporte a essa funcionalidade, são elas: GP\_SUPPORT\_OCCURRENCE, GP\_OCCURRENCE\_STATUS e GP\_LOG\_SUPPORT\_MESSAGE.

Quando o cliente for abrir um caso de suporte no ambiente, os detalhes iniciais do mesmo devem ser armazenados na tabela GP\_SUPPORT\_OCCURRENCE, tabela criada para armazenar as ocorrências de suporte. Na tabela, é permitido armazenar um código identificador, título resumido da ocorrência, descrição completa, código do projeto ao qual ela está relacionada (caso exista), código do usuário que criou a ocorrência, código do status da ocorrência, código do usuário alocado para atendimento da ocorrência, código do arquivo anexado pelo cliente (caso exista), código do cliente responsável pela ocorrência de suporte, data de criação e data de fechamento do caso, quando o atendimento é encerrado.

Para armazenar o código do projeto, foi criada uma relação com a tabela GP\_PROJECT. Para armazenar o código do usuário que criou o caso e o usuário alocado para o caso, foi criada uma relação com a tabela GP\_USER. Para armazenar o código do arquivo armazenado pelo cliente, foi criada uma relação com a tabela GP\_FILE. Para armazenar o código do cliente responsável pela ocorrência de suporte, foi criada uma relação com a tabela GP\_COMPANY.

Para armazenar o código do status da ocorrência de suporte, foi criada uma relação com a tabela GP\_OCCURRENCE\_STATUS. A tabela então permite armazenar o código do status de ocorrência, nome e descrição. Por default, foram criados cinco exemplos de status de ocorrências, conforme a Tabela 10.

Tabela 10 Registros default de status de ocorrências

ID	Status de Ocorrências	Descrição
1	Aberto	A ocorrência acaba de ser cadastrada
2	Em Análise	Analisando uma solução para a ocorrência
3	Em Desenvolvimento	A solução para o problema está sendo desenvolvida
4	Em Teste	A solução para o problema está sendo testada
5	Finalizado	O problema foi corrigido no cliente

Os registros default da Tabela 10 podem ser modificados conforme a necessidade do usuário. Existe também a possibilidade de adicionar outros registros para tratar outros status de ocorrência.

### 5.1.11 Configurações

A funcionalidade de configurações dá suporte à necessidade de diferentes tipos de configurações do ambiente. A estrutura criada foi feita baseando na ideia de cada configuração ser expressão em forma de um registro, um parâmetro. A estrutura da Figura 59 mostra a tabela criada para essa funcionalidade.

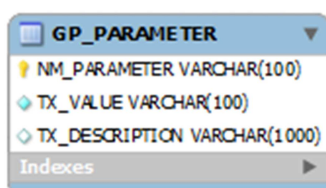


Figura 59 Tabela da funcionalidade de Configurações

Como mostra na Figura 59, foi criada a tabela GP\_PARAMETER, que armazenará registro, ou seja, parâmetros de configuração do sistema. É possível armazenar um nome do parâmetro, que é chave primária, o valor do parâmetro e a descrição.

Tomando como exemplo a definição de uma configuração do sistema de idiomas. Para adicionar essa configuração basta inserir um registro na tabela com as definições necessárias, definindo um nome para o parâmetro, um valor e a sua descrição. Esse registro seria verificado pelo sistema quando necessário e seria retornado o valor armazenado. Na Tabela 11 segue um exemplo de parâmetro a ser criado para realizar o tratamento de idiomas.

Tabela 11 Exemplo de parâmetro da tabela de configuração

Parâmetro	Valor	Descrição
'SystemLanguage'	'pt-br'	'Parâmetro que indica o idioma do sistema. Para o valor <i>pt-br</i> o idioma é o português do Brasil. Para o valor <i>en</i> o idioma é o inglês. Para o valor 'es' o idioma é espanhol.'

Vários parâmetros devem ser inseridos para utilização do sistema. Os mesmos devem ser definidos de acordo com a necessidade do desenvolvimento do ambiente.

## 6 RESULTADOS

O novo modelo de banco de dados para um ambiente de gerência de projetos pode ser definido como principal resultado obtido durante o desenvolvimento deste trabalho. O modelo de banco de dados criado teve como preocupação vários aspectos importantes durante o seu desenvolvimento, preocupações essas direcionadas ao objetivo de facilitar a utilização do usuário no contexto de desenvolvimento e gerenciamento de projetos e afins.

Durante a sua criação, um cuidado importante adotado foi o de realizar a documentação de todos os objetos criados, tais como comentários de tabelas, colunas, relações entre tabelas, dentre outros. Tal preocupação se deu ao fato de que o desenvolvimento da nova ferramenta de gerência de projetos acontecerá em um momento posterior ao da modelagem do banco de dados.

Vale ressaltar que o modelo de banco de dados criado foi uma primeira versão para a nova ferramenta de gerência de projetos e, por esse motivo, novas ideias podem surgir e, conseqüentemente, funcionalidades podem ser incrementadas ou excluídas.

Para dar suporte a eventuais alterações pode ser necessário alterar a lógica ou os objetos do modelo do banco de dados e, por isso, foi criado um dicionário de dados, incluso no apêndice desse trabalho, para descrever todas as características dos objetos criados nesse novo modelo.

O novo modelo de banco de dados pode ser vantajoso quanto aos outros já existentes pelo fato dele ser modelado de acordo com as principais características de três ambientes de gerência de projetos bastante utilizado no mercado.

Além disso, como não há nenhuma interface ou aplicativo em cima do banco de dados, é possível modifica-lo à medida que a nova ferramenta de gerência de projetos é desenvolvida, facilitando o retrabalho e a implementação.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Gerenciar projetos em geral é uma difícil missão para os gerentes de projetos, uma vez que vários fatores estão envolvidos nesse processo. Além disso, muitos desses fatores são imprevisíveis e complexos. Por esse motivo várias ferramentas foram e estão sendo desenvolvidas para que possam ser utilizadas pelos gerentes de projetos para tentar gerenciar e prever certos fatores. Os ambientes de gerência de projetos têm se destacado nesse contexto e se consolidam cada dia mais como uma grande ferramenta para auxiliar o gerente de projetos em seu cotidiano.

A documentação é fundamental em um projeto de desenvolvimento de um modelo de banco de dados, pois ela permite ao desenvolvedor total conhecimento de sua estrutura e complexidade. Uma manutenção ou modificação de um modelo de banco de dados documentado é mais eficiente e eficaz do que um modelo de banco de dados sem documentação.

Os objetivos do trabalho foram alcançados com a criação do modelo de banco de dados e sua documentação. Levar em consideração os ambientes de gerência de projetos já utilizados foi de extrema importância para inserção no contexto do tema e também no conhecimento quanto à estrutura e funcionamento desses ambientes.

Quanto à limitação da pesquisa, é válido salientar que o modelo de banco de dados desenvolvido foi oriundo de uma análise pessoal, mesmo que apoiada por alguns pilares. Além disso, a sintaxe do modelo desenvolvido foi feito baseado no Sistema Gerenciador de Banco de Dados MySQL, ponto esse que pode ser superado com várias ferramentas disponíveis de migração de diferentes Sistemas Gerenciadores de Banco de Dados.

Como trabalhos futuros, a expectativa é que o modelo de banco de dados criado nesse trabalho possa servir como base para outros projetos que

eventualmente desenvolva uma interface e uma lógica de programação para um novo ambiente de gerenciamento de projetos.

Outro ponto que pode ser considerado futuramente é a criação de novas funcionalidades para o modelo desenvolvido, levando em considerações outros tipos de observações e análises de ambientes. Nesse ponto, saliento uma abordagem com gerentes de projetos atuantes na área para opinião de melhorias ou até mesmo o acompanhamento de um projeto real em uma determinada empresa de software com o objetivo de detectar necessidades não levantadas nesse trabalho.

## 8 REFERÊNCIAS BIBLIOGRÁFICAS

PHILLIPS, J. *Gerência de Projetos de Tecnologia da Informação*, Editora Campus, 2003, 16p.

DOTPROJECT, 2011. Disponível em: <<http://www.dotproject.net>>. Acesso em: 09 nov. 2011.

NETOFFICE, 2011. Disponível em: <<http://sourceforge.net/projects/netoffice>>. Acesso em: 19 nov. 2011.

DATE, C. J. *Introdução à Sistemas de Banco de Dados*, Elsevier, 2003.

SILBERSCHATZ, A.; KORTH, H.F.; SUDARSHAN, S. *Sistema de banco de dados*, Editora Campus, 5ª edição, 2006, 1p.

ELMASRI, R.; NAVATHE, S.B. *Sistemas de Banco de Dados*, Pearson Education, 6ª edição, 2011, 39p.

TEOREY, T.; LIGHTSTONE, S.; NADEAU, T. *Projeto e Modelagem de Banco de Dados*, Editora Campus, 4ª Edição, 2006, 9p.

MACHADO, F.; ABREU, M. *Projeto de Banco de Dados: Uma Visão Prática*, Editora Érica, 16ª Edição, 2009, 47p.

PMBOK, *Project Management Body of Knowledge (PMBOK)*, PMI, 4ª Edição, 2008

OCHNER, J. *Gerência de Projetos: Uma comparação entre o PMBOK e XPM*, UFLA, 2006

FRAME, J. Davison. *Managing projects in organization*, São Francisco – Jossey – Bass Inc., 1995

SCHERRER, Celson Rohr. *O sucesso em projetos de TI e a terceirização*. Rio de Janeiro, Março de 2009

OSI, 2011. Disponível em: <<http://www.opensource.org/docs/definition.php>>. Acesso em: 19 nov. 2011.



WAINER, J. *Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação*, Sociedade Brasileira de Computação e Editora PUC-Rio, 2007. P. 221-262

CODAS, M. M. B. *Gerência de Projetos: Uma Reflexão Histórica*, Revista Administração de Empresas, Rio de Janeiro, jan/mar 1987. P. 33-37

CORREIA, J. SANTOS, M. *UML-Implementação com Bancos de Dados Relacionais*. Instituto Superior Politécnico de Gaya, 2002.

FOWLER, M. *UML Essencial: um breve guia para a linguagem padrão de modelagem de objetos* - 3.ed. - Porto Alegre : Bookman, 2005.

## APÊNDICE A

Para fins de documentação, foi criado um dicionário de dados para o novo modelo de banco de dados desenvolvido durante o trabalho. São 29 tabelas no total.

No campo EXTRA, considerar as seguintes nomenclaturas como:

- NN: Campo definido como not null (não nulo).
- N: Campo definido como null (nulo).
- AI: Campo auto incremento.

### 1. GP\_PROJECT: Tabela para armazenar projetos

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_PROJECT	INT	AI, NN	Código do projeto
NM_PROJECT	VARCHAR(60)	NN	Nome do projeto
NM_SHORT_PROJECT	VARCHAR(20)	NN	Código simbólico do projeto
DT_INSERT	DATETIME	N	Data de inserção do registro na tabela
DT_START_PREV	DATETIME	N	Data prevista de início do projeto
DT_END_PRE	DATETIME	N	Data final prevista do projeto
DT_START	DATETIME	N	Data inicial do projeto
DT_END	DATETIME	N	Data final do projeto
ID_MANAGER	INT	N	Código do usuário gerente do projeto
ID_PROJECT_CATEGORY	INT	N	Código da categoria do projeto
ID_CLIENT	INT	N	Código do cliente do projeto
QT_COST_PREV	DECIMAL(15,2)	N	Custo previsto de desenvolvimento do projeto
QT_COST_REAL	DECIMAL(15,2)	N	Custo real de desenvolvimento do projeto
QT_SALE_COST	DECIMAL(15,2)	N	Custo de venda do projeto ao cliente
TX_DESCRIPTION	VARCHAR(1046)	N	Observações do projeto
ID_SECTOR	INT	N	Setor responsável pelo projeto
ID_PROJECT_STATUS	INT	-	Código do status do projeto

## 2. GP\_USER: Tabela para armazenar usuários

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_USER	INT	AI, NN	Código do usuário
NM_USER	VARCHAR(60)	NN	Nome do usuário
TX_PASSWORD	VARCHAR(100)	N	Password do usuário
NM_FIRST_NAME	VARCHAR(45)	NN	Primeiro nome do usuário
NM_SECOND_NAME	VARCHAR(150)	NN	Segundo nome do usuário
DT_BIRTHDAY	DATE	N	Data de nascimento do usuário
DT_INSERT	DATETIME	N	Data de inserção do usuário
ID_SECTOR	INT	N	Código do departamento, setor do funcionário
QT_SALARY	DECIMAL(10,2)	N	Salário do funcionário
ID_USER_FUNCTION	INT	NN	Código da função do usuário
CH_PERMISSION_PROJECT	INT	NN	Indica se o usuário tem permissão de criar projetos (1) ou não (0)
CH_PERMISSION_USER	INT	NN	Indica se o usuário tem permissão de adicionar usuários (1) ou não (0)
CH_PERMISSION_SYSTEM	INT	NN	Indica se o usuário tem permissão de ajustar configurações do sistema (1) ou não (0)
DT_DELETE	DATETIME	N	Data de deleção do funcionário

## 3. GP\_SECTOR: Tabela para armazenar setores

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_SECTOR	INT	AI, NN	Código do setor
NM_SECTOR	VARCHAR(60)	NN	Nome do setor
ID_MANAGER	INT	N	Código do gerente do setor
QT_USERS	INT	N	Quantidade de funcionários existentes no módulo
DT_INSERT	DATETIME	N	Data de inserção do setor
DT_DELETE	DATETIME	N	Data de deleção do setor

**4. GP\_COMPANY:** Tabela para armazenar organizações, sejam elas clientes, fornecedores, módulos internos, entre outros.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_COMPANY	INT	AI, NN	Código da organização (PK)
NM_COMPANY	VARCHAR(60)	NN	Nome da organização
CH_TYPE_PERSON	INT	NN	Indica se o cliente é pessoa física (1) ou jurídica(0)
ID_COMPANY_TYPE	INT	NN	Código do tipo de organização.
NM_CITY	VARCHAR(60)	N	Cidade da organização
NM_STATE	VARCHAR(60)	N	Estado UF da organização
NM_COUNTRY	VARCHAR(60)	N	País da organização
TX_ADREESS	VARCHAR(250)	N	Endereço da organização
TX_FONE	VARCHAR(45)	N	Número do telefone da organização
TX_MAIL	VARCHAR(100)	N	E-mail da organização
TX_WEBSITE	VARCHAR(100)	N	Endereço URL do website da organização
DT_INSERT	DATETIME	N	Data em que a organização foi cadastrada
DT_DELETE	DATETIME	N	Data de deleção da organização

**5. GP\_TASK:** Tabela para armazenar tarefas dos projetos

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_TASK	INT	AI, NN	Código da tarefa
NM_TASK	VARCHAR(60)	NN	Nome da tarefa
ID_PROJECT	INT	N	Código do projeto da tarefa
DT_INSERT	DATETIME	N	Data de criação da tarefa
DT_START	DATETIME	N	Data real de início da tarefa
DT_END	DATETIME	N	Data real de término da tarefa
DT_START_PREV	DATETIME	N	Data prevista de início da tarefa
DT_END_PREV	DATETIME	N	Data prevista de término da tarefa
QT_PREV_HOUR	DECIMAL(8,2)	N	Quantidade prevista de horas para a tarefa
QT_HOUR	DECIMAL(8,2)	N	Quantidade de horas gastas na tarefa
QT_PROGRESS	INT	N	Porcentagem de finalização da tarefa
CH_INDIVIDUAL_TASK	INT	N	Indica se a tarefa é individual(1) ou não

			(0).
ID_WORKER	INT	N	Código do funcionário a qual a tarefa foi delegada
ID_PROJECT_PHASE	INT	NN	Código da fase do projeto da tarefa
ID_SUPPORT_OCCURRENCE	INT	N	Código da ocorrência de suporte, caso a tarefa seja relacionada à correção de alguma ocorrência.

**6. GP\_USER\_FUNCTION:** Tabela para adicionar os cargos dos usuários

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_USER_FUNCTION	INT	AI, NN	Código da função do usuário
NM_USER_FUNCTION	VARCHAR(60)	NN	Nome da função do usuário
QT_MIN_SALARY	DECIMAL(15,2)	N	Valor mínimo do salário da função
QT_MAX_SALARY	DECIMAL(15,2)	N	Valor máximo do salário da função
TX_DESCRIPTION	VARCHAR(1000)	N	Descrição da função do usuário
DT_DELETE	DATETIME	N	Data de deleção

**7. GP\_COMPANY\_TYPE:** Tabela para armazenar o tipo de organização

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_COMPANY_TYPE	INT	AI, NN	Código do tipo de organização
NM_COMPANY_TYPE	VARCHAR(60)	NN	Nome do tipo de organização
TX_DESCRIPTION	VARCHAR(250)	N	Descrição do tipo de companhia

**8. GP\_LOG\_TASK:** Tabela para armazenar o log de tarefas

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_LOG_TASK	INT	AI, NN	Código do log de lançamento de tarefa
ID_TASK	INT	NN	Código da tarefa. FK da tabela de tarefas.
ID_WORKER	INT	NN	Código do usuário que realizou a tarefa
DT_START	DATETIME	N	Data de início do lançamento da tarefa

DT_END	DATETIME	N	Data de fim do lançamento da tarefa
TX_DESCRIPTION	VARCHAR(256)	N	Observação da atividade realizada.
QT_PROGRESS	INT	N	Percentual de conclusão da tarefa após o lançamento.
DT_ADJUST	DATETIME	N	Data de edição da tarefa
ID_USER_ADJUST	INT	N	Código da pessoa que realizou o ajuste na tarefa
ID_WORK_PLACE	INT	N	Tipo do lançamento de horas

**9. GP\_PROJECT\_CATEGORY:** Tabela para armazenar as categorias de projetos.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_PROJECT_CATEGORY	INT	AI, NN	Código da categoria do projeto
NM_PROJECT_CATEGORY	VARCHAR(60)	NN	Nome da categoria de projeto
DT_DELETE	DATETIME	N	Data de deleção
TX_DESCRIPTION	VARCHAR(256)	N	Descrição da categoria do projeto

**10. GP\_WORK\_PLACE:** Tabela para armazenar onde o trabalho foi realizado pelo funcionário.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_WORK_PLACE	INT	AI, NN	Código do local do trabalho realizado
NM_WORK_PLACE	VARCHAR(60)	NN	Nome do local do trabalho realizado
TX_DESCRIPTION	VARCHAR(250)	N	Observação do tipo de lançamento

**11. GP\_PROJECT\_STATUS:** Tabela para armazenar os status dos projetos.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_PROJECT_STATUS	INT	AI, NN	Código do status do projeto
NM_PROJECT_STATUS	VARCHAR(60)	NN	Nome do status do projeto
TX_DESCRIPTION	VARCHAR(250)	N	Descrição do status do projeto

**12. GP\_EVENT:** Tabela para armazenar os eventos cadastrados pelos usuários.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_EVENT	INT	AI, NN	Código do evento
NM_EVENT	VARCHAR(60)	NN	Nome do evento
DT_START	DATETIME	N	Data de início do evento
DT_END	DATETIME	N	Data de fim do evento
TX_PLACE	VARCHAR(60)	N	Local do evento
TX_DESCRIPTION	VARCHAR(250)	N	Descrição do evento
CH_GENERATE_ALARM	INT	N	Indica se o evento vai gerar (1) ou não (0) alarme para os usuários participantes
ID_PROJECT	INT	N	Código do projeto do evento
ID_COMPANY	INT	N	Indica o código da organização participante do evento
ID_EVENT_TYPE	INT	N	Código do tipo do evento
ID_LEADER	INT	N	Código do usuário responsável pelo evento

**13. GP\_EVENT\_TYPE:** Tabela para armazenar os tipos de eventos.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_EVENT_TYPE	INT	AI, NN	Código do tipo de evento
NM_EVENT_TYPE	VARCHAR(60)	NN	Nome do tipo de evento
TX_DESCRIPTION	VARCHAR(250)	N	Descrição do tipo de evento

**14. GP\_CONTACT:** Tabela para armazenar dados de contatos.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_CONTACT	INT	AI, NN	Código do contato
NM_FIRST_NAME	VARCHAR(60)	NN	Primeiro nome do contato
NM_SECOND_NAME	VARCHAR(100)	N	Segundo nome do contato
ID_COMPANY	INT	N	Indica se o contato está relacionado a alguma organização já cadastrada no

			sistema
CH_INDIVIDUAL_CONTACT	INT	NN	Indica se a visualização do contato é apenas para quem cadastrou (1) ou todos podem ver (0)
TX_EMAIL	VARCHAR(100)	N	E-mail do contato
TX_PERSONAL_PHONE	VARCHAR(20)	N	Numero do telefone pessoal do contato
TX_JOB_PHONE	VARCHAR(20)	N	Numero do telefone de serviço do contato
TX_HOME_PHONE	VARCHAR(20)	N	Numero do telefone da casa do contato
TX_ADREESS	VARCHAR(200)	N	Endereço do contato
NM_CITY	VARCHAR(45)	N	Nome da cidade do contato
NM_STATE	VARCHAR(45)	N	Nome do estado do contato
NM_COUNTRY	VARCHAR(45)	N	Nome do país do contato
NM_JOB	VARCHAR(45)	N	Profissão do contato
DT_BIRTHDAY	DATE	N	Data de nascimento do contato
TX_DESCRIPTION	VARCHAR(1000)	N	Outras observações do contato

**15. GP\_PROJECT\_GOAL:** Tabela para armazenar metas e objetivos de projetos.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_PROJECT_GOAL	INT	AI, NN	Código da meta de projeto
NM_PROJECT_GOAL	VARCHAR(100)	NN	Nome da meta de projeto
DT_FINISH	DATE	N	Data para alcance da meta estabelecida
ID_PROJECT	INT	NN	Código do projeto a qual a meta pertence
TX_DESCRIPTION	VARCHAR(250)	N	Observação e descrição da meta
DT_DELETE	DATE	N	Data de deleção da meta
TX_REASON_DELETE	VARCHAR(250)	N	Motivo de deleção da meta, caso necessário



**16. GP\_EVENT\_USER:** Tabela para armazenar associações de eventos com usuários. Ela permite armazenar o histórico de participação dos usuários nos eventos já realizados.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_EVENT	INT	NN	Código do evento
ID_USER	INT	NN	Código do usuário

**17. GP\_PROJECT\_PHASE:** Tabela para armazenar as fases de um projeto.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_PROJECT_PHASE	INT	AI, NN	Código da fase do projeto
NM_PROJECT_PHASE	VARCHAR(60)	NN	Nome da fase do projeto
NU_ORDER	VARCHAR(45)	N	Ordem da fase no ciclo de vida do projeto
TX_DESCRIPTION	VARCHAR(1000)	N	Descrição da fase do projeto

**18. GP\_LOG\_PROJECT\_COST:** Tabela para armazenar custos do projeto.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_LOG_PROJECT_COST	INT	AI, NN	Código do log de custo do projeto
ID_PROJECT	INT	NN	Código do projeto do custo
NM_PROJECT_COST	VARCHAR(60)	NN	Nome do custo do projeto
ID_PROJECT_COST_CATEGORY	INT	NN	Código da categoria de custo de projeto
ID_WORKER	INT	NN	Código do funcionário que realizou a despesa
ID_USER_INSERT	INT	NN	Código do usuário que realizou a inserção do custo de projeto
DT_EVENT	DATE	N	Data em que ocorreu o custo
TX_DESCRIPTION	VARCHAR(1000)	N	Descrição do custo de projeto

**19. GP\_PROJECT\_COST\_CATEGORY:** Tabela para armazenar categorias de custo de projetos.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_PROJECT_COST_CATEGORY	INT	AI, NN	Código da categoria de custo de projeto
NM_PROJECT_COST_CATEGORY	VARCHAR(60)	NN	Nome da categoria de custo de projeto
TX_DESCRIPTION	VARCHAR(1000)	N	Descrição da categoria de custo de projeto

**20. GP\_FILE:** Tabela para armazenar informações de arquivos do ambiente.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_FILE	INT	AI, NN	Código do arquivo
NM_FILE	VARCHAR(45)	NN	Nome do arquivo
ID_PROJECT	INT	N	Código do projeto do arquivo
DT_INSERT	DATETIME	N	Data de inserção do arquivo
NM_FILE_EXTENSION	VARCHAR(10)	N	Extensão do arquivo
QT_SIZE_KB	DECIMAL(20,2)	N	Tamanho do arquivo em KB
TX_VERSION	VARCHAR(15)	N	Versão do arquivo
TX_DESCRIPTION	VARCHAR(250)	N	Descrição do arquivo

**21. GP\_PROJECT\_MEMBER:** Tabela de associação de usuários, permissões e projetos.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_PROJECT	INT	NN	Código do projeto
ID_USER	INT	NN	Código do usuário
CH_VIEW	INT	NN	Indica se o usuário terá permissão para visualizar o projeto (1) ou não (0).
CH_EDIT	INT	NN	Indica se o usuário terá permissão para editar o projeto (1) ou não (0).

CH_ADD	INT	NN	Indica se o usuário terá permissão para criar objetos no projeto (1) ou não (0).
CH_DELETE	INT	NN	Indica se o usuário terá permissão para deletar objetos no projeto (1) ou não (0).

**22. GP\_FORUM:** Tabela para armazenar fóruns do ambiente.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_FORUM	INT	AI, NN	Código do fórum
NM_FORUM	VARCHAR(60)	NN	Nome do fórum de discussão
ID_PROJECT	INT	N	Código do projeto ao qual o fórum pertence
TX_DESCRIPTION	VARCHAR(500)	N	Descrição do fórum de discussão
DT_CREATE	DATETIME	NN	Data de criação do fórum

**23. GP\_FORUM\_TOPIC:** Tabela para armazenar tópicos dos fóruns de discussão.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_FORUM_TOPIC	INT	AI, NN	Código do tópico de fórum
NM_FORUM_TOPIC	VARCHAR(45)	NN	Nome do tópico do fórum
ID_FORUM	INT	NN	Código do fórum ao qual o tópico pertence
ID_USER	INT	NN	Usuário que criou o tópico
TX_DESCRIPTION	VARCHAR(1000)	N	Descrição do tópico
DT_CREATE	DATETIME	NN	Data de criação do tópico

**24. GP\_SUPPORT\_OCCURRENCE:** Tabela para armazenar ocorrências de suporte, tanto internas quanto externas.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_SUPPORT_OCCURRENCE	INT	AI, NN	Código da ocorrência de suporte
NM_SUPPORT_OCCURRENCE	VARCHAR(60)	NN	Nome da ocorrência de suporte

TX_DESCRIPTION	VARCHAR(1000)	NN	Descrição do problema ocorrido
ID_PROJECT	INT	N	Código do projeto ao qual a ocorrência de suporte está relacionada
ID_REPORT_USER	INT	NN	Usuário que reportou a ocorrência de suporte
ID_OCCURRENCE_STATUS	INT	N	Indica o status da ocorrência
ID_ALLOCATED_USER	INT	N	Código do usuário interno alocado para resolução da ocorrência de suporte
ID_FILE	INT	N	Código do arquivo em anexo, caso exista
ID_CLIENT	INT	N	Código do cliente que cadastrou a ocorrência de suporte
DT_CREATE	DATETIME	N	Data em que o caso foi aberto
DT_CLOSE	DATETIME	N	Data em que o caso foi fechado

**25. GP\_LOG\_FORUM\_MESSAGE:** Tabela para armazenar o histórico de mensagens dos fóruns de discussão.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_LOG_FORUM_MESSAGE	INT	AI, NN	Código do log da mensagem do fórum
ID_FORUM_TOPIC	INT	NN	Código do tópico do fórum
ID_USER	INT	NN	Código do usuário
TX_ANSWER	VARCHAR(1000)	NN	Texto com a resposta do usuário
DT_MESSAGE	DATETIME	NN	Data da mensagem do fórum

**26. GP\_OCCURRENCE\_STATUS:** Tabela para armazenar os status das ocorrências de suporte.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_OCCURRENCE_STATUS	INT	AI, NN	Código do status da ocorrência
NM_OCCURRENCE_STATUS	VARCHAR(45)	NN	Nome do status da ocorrência de suporte
TX_DESCRIPTION	VARCHAR(300)	N	Descrição do status da ocorrência de suporte

**27. GP\_LOG\_SUPPORT\_MESSAGE:** Tabela para armazenar o log das mensagens de ocorrência de suporte.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_LOG_SUPPORT_MESSAGE	INT	AI, NN	Código do log da mensagem de atendimento ao suporte
ID_SUPPORT_OCCURRENCE	INT	NN	Código da ocorrência de suporte
TX_DESCRIPTION	VARCHAR(500)	NN	Descrição da operação feita
ID_USER	INT	NN	Usuário que postou a mensagem
DT_MESSAGE	DATETIME	NN	Data da mensagem

**28. GP\_PARAMETER:** Tabela para armazenar os parâmetros e as configurações do ambiente.

CAMPO	TIPO	EXTRA	COMENTARIOS
NM_PARAMETER	VARCHAR(100)	NN	Nome do parâmetro. PK da tabela.
TX_VALUE	VARCHAR(100)	NN	Valor do parâmetro
TX_DESCRIPTION	VARCHAR(1000)	N	Descrição do parâmetro

**29. GP\_LOG\_PROJECT\_STATUS:** Tabela para armazenar histórico de status de projetos.

CAMPO	TIPO	EXTRA	COMENTARIOS
ID_LOG_PROJECT_STATUS	INT	AI, NN	Código do log de status de projeto
ID_PROJECT	INT	NN	Código do projeto
ID_PROJECT_STATUS	INT	NN	Código do status de projeto
DT_CHANGE	DATETIME	NN	Data de alteração do status do projeto
TX_DESCRIPTION	VARCHAR(500)	N	Descrição do motivo de mudança do status do projeto

## **APÊNDICE B**

O modelo do banco de dados resultante desse trabalho é demonstrado abaixo. O modelo foi desenvolvido na ferramenta MySQL Workbench e demonstra todos os objetos do banco de dados, incluindo suas tabelas e relações, bem como seus campos.

