



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
ESCOLA DE INFORMÁTICA APLICADA

Uma ferramenta de cálculo de esforço para projetos de desenvolvimento de software baseada no método de Pontos de Função.

Casimiro Conde Marco Neto

**Orientador**

Márcio de Oliveira Barros

Rio de Janeiro, RJ – Brasil

2013

Uma ferramenta de cálculo de esforço para projetos de desenvolvimento de software baseada no método de Pontos de Função.

Casimiro Conde

Os autores desse projeto autorizam a ESCOLA DE INFORMÁTICA APLICADA da UNIRIO a divulgá-lo, no todo ou em parte, resguardando os direitos autorais conforme legislação vigente.

Rio de Janeiro, \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

---

Casimiro Conde Marco Neto

Aprovada por:

---

Prof. Márcio de Oliveira Barros, DSc (UNIRIO)

---

Prof. Leila Cristina V. de Andrade, DSc (UNIRIO)

---

Prof. Alexandre Luis Correa, DSc (UNIRIO)

Rio de Janeiro, RJ – Brasil

2013

## **Agradecimentos**

Ao professor Márcio Barros pela orientação, empenho, apoio e continua presença em todas as etapas de desenvolvimento, fornecendo toda a ajuda e materiais necessários para a realização desse projeto.

A professora Leila Andrade por ter acompanhado meu amadurecimento como aluno e pessoa desde os primeiros dias de vida acadêmica e ter aceitado participar da banca.

A todos os colegas e pessoas especiais que me acompanharam e incentivaram durante toda a minha jornada da graduação.

Ao professor Alexandre Correa por ter aceitado o convite para participar da banca.

Aos amigos e família que forneceram toda a estrutura e incentivos necessários para o sucesso da minha formação acadêmica e continuam a me acompanhar durante todos os sucessos e derrotas da vida.

## Resumo

Esse projeto apresenta uma ferramenta para dimensionamento em Pontos de Função para projetos de desenvolvimento de software, se baseando nos requisitos do software em questão.

Para a execução desse projeto foi necessária pesquisa e o entendimento sobre os conceitos de gerenciamento de projetos e a técnica de contagem de Pontos de Função. Além dessas atividades, também foram executadas as outras fases de um projeto de desenvolvimento de software, incluindo a análise e definição de requisitos, modelagem do sistema a ser desenvolvido, construção do código da aplicação e testes do sistema implementado.

Para o desenvolvimento do projeto foram estudadas e utilizadas as tecnologias Google App Engine e o NoSQL.

**Palavras-chaves:** Gerência de Projetos, Estimativa de Esforço, Pontos de Função, Sistemas de Informação.

## **Abstract**

This project introduces a tool that calculates the size for software development projects in function points, based on the software requirements.

To implement this project, I researched and understood concepts related to project management and the function points counting technique. Besides that, other phases of a software development project were also performed, such as requirement analysis, system modeling, coding and testing the application.

For the development were used the technologies Google App Engine and NoSQL.

**Keywords:** Project Management, Effort Estimation, Function Points, Information Systems.

# Conteúdo

<b>1 Introdução</b>	<b>1</b>
1.1 - Motivação	2
1.2 - Objetivos	3
1.3 - Organização do Texto	3
<b>2 Gerência de Projetos</b>	<b>5</b>
2.1 - Definição de Projeto	5
2.2 - Áreas de Gerenciamento de Projeto	5
2.3 - Gerenciamento de Tempo	7
2.4 - Análise de Pontos por Função	7
2.4.1 - A necessidade de medição	8
2.4.2 - Objetivos da contagem por Pontos de Função	9
2.4.3 - Conceitos de Pontos de Função	9
2.4.4 - Etapas do processo de avaliação	10
2.4.5 - Contagem de Pontos de Função de Dados	12
2.4.6 - Contagem de Pontos de Função Transacionais	13
2.4.7 - Utilização dos Pontos de Função para Geração de Métricas	16
2.5 – Considerações Finais	16
<b>3 Tecnologias e Requisitos do Sistema</b>	<b>17</b>
3.1 - Requisitos do Sistema	17
3.2 - Modelo Conceitual	19
3.3 - Google App Engine	20
3.4 - NoSQL	23
3.5 - Ambiente de desenvolvimento e linguagem	26
3.6 - Considerações Finais	27
<b>4 Sistema de cálculo de esforço para projetos de software baseado em Pontos de Função</b>	<b>28</b>
4.1 - Cadastro de Usuário	28

4.2 - Funcionalidades do Sistema	30
4.3 Considerações Finais	35
<b>5 Conclusões</b>	<b>37</b>
5.1 - Observações Finais	37
5.2 - Evoluções Futuras	38
5.3 - Limitações do Projeto	38
<b>Referências Bibliográficas</b>	<b>39</b>

## Índice de Tabelas

Tabela 1 – Nível de Complexidade de um ILF.....	13
Tabela 2 – Contribuição de cada ILF para a contagem de Pontos de Função.....	13
Tabela 3 – Nível de Complexidade de um EIF .....	13
Tabela 4 – Contribuição de cada EIF para a contagem de Pontos de Função .....	13
Tabela 5 – Nível de complexidade de um EI .....	14
Tabela 6 – Contribuição de cada EI para a contagem de Pontos de Função .....	14
Tabela 7 – Nível de complexidade de um EO .....	15
Tabela 8 – Contribuição de cada EO para a contagem de Pontos de Função.....	15
Tabela 9 – Nível de Complexidade de um EQ .....	15
Tabela 10 – Contribuição de cada EQ para a contagem de Pontos de Função.....	15

## Índice de Figuras

Figura 1 – Diagrama de Casos de Uso do Sistema .....	17
Figura 2 – Diagrama de Classes do Sistema .....	20
Figura 3 – Comparação entre Banco Orientado a Linhas e Orientado a Colunas .....	25
Figura 4 – Tela de Login .....	28
Figura 5 – Tela de Cadastro de Usuário .....	29
Figura 6 – Tela de Recuperação de Senha.....	29
Figura 7 – Tela de Reset de Senha .....	30
Figura 8 – Tela Inicial do Sistema.....	31
Figura 9 – Tela de Informações do Projeto .....	32
Figura 10 – Tela de Informações da Função de Dados .....	33
Figura 11 – Tela de Informações da Função Transacional .....	33
Figura 12 – Demonstração do número de pontos de função de um projeto .....	34
Figura 13 – Tela de Exportação do projeto em formato XML.....	35
Figura 14 – Tela de Edição de Informações.....	35



# 1 Introdução

Nos dias atuais, empresas de todos os portes e de diversos seguimentos segmentos estão se rendendo cada vez mais à tecnologia e aos seus benefícios, estão automatizando tarefas e processos a ponto de já existirem diversas empresas que dependem fortemente da tecnologia para sobreviver, e que sem ela não poderiam manter seus serviços e atividades básicas em funcionamento, tais como empresas de telecomunicação e hospedagem web. Além delas, existem empresas que são vendedoras e desenvolvedoras de novas tecnologias, ou seja, sobrevivem da criação e inovação, como empresas de eletrônicos, celulares, automobilística, entre outros tantos ramos dependentes da inovação tecnológica.

Entre os inúmeros tipos de tecnologia existentes no mundo, uma das mais importantes, senão a mais importante, é a tecnologia computacional, aquela que faz os computadores e os sistemas funcionarem. Sistemas esses que são base para muitos dos serviços hoje consumidos por nós. Sistemas que mantêm nossas informações seguras e são base de muitos serviços básicos que utilizamos no dia-a-dia.

Esses sistemas, chamados Sistemas de Informação, são construídos e evoluídos através de projetos, projetos esses que têm como objetivo o desenvolvimento de uma solução para atender a uma necessidade de negócio do cliente ou usuário do sistema. Sendo assim, esses projetos tornaram-se fundamentais para a evolução de diversas empresas. Através desses projetos as empresas podem suprir necessidades e criar instrumentos para que os objetivos estratégicos traçados sejam alcançados, novos produtos sejam desenvolvidos e para promover o crescimento e valorização da instituição.

Uma das partes mais importantes para o projeto é a estimativa do esforço que será despendido para sua execução. Essa estimativa é fruto de planejamento e da modelagem do projeto. Uma das técnicas mais comuns utilizadas é o Ponto por Função, uma medida de esforço que possibilita o planejamento das atividades e da duração estimada, gerando assim uma perspectiva do andamento do projeto.

## 1.1 - Motivação

A oportunidade de construir uma ferramenta, na qual empresas e gerentes de projeto possam executar o cálculo da dimensão de um projeto em pontos de função através de uma descrição detalhada dos requisitos, mantendo a memória de cálculo detalhada por função e as possibilidades de expansão e que essa ferramenta possui são as motivações para a elaboração e desenvolvimento desse projeto. Somado a isso existe o meu interesse pessoal e profissional na Gerência de Projetos de Software, sendo o meu principal objetivo de carreira me tornar um profissional qualificado e reconhecido na disciplina de Gerenciamento de Projetos e suas vertentes. Sendo assim, identifiquei neste projeto a oportunidade de me aprofundar em uma das áreas de conhecimento mais importantes para um projeto: o gerenciamento de tempo.

Além das motivações anteriormente apresentadas, a possibilidade de fortalecer minhas habilidades de desenvolvimento de software também foi um grande atrativo para a escolha desse tema para o projeto de conclusão de curso. Até o início desse projeto, o contato do autor com as técnicas de desenvolvimento não era maior do que o necessário para a elaboração dos trabalhos nas disciplinas de desenvolvimento e o contato eventual por curiosidade pessoal, como o desenvolvimento para plataformas móveis. Ou seja, o autor desse projeto nunca esteve profissionalmente no papel de um desenvolvedor e nem envolvido em nenhum projeto extra curricular nessa posição. Dessa forma, o contato com esse “lado da moeda” do mundo de projetos de software tornou a proposta ainda mais interessante.

Por último, o contato com tecnologias de desenvolvimento web mundialmente conhecidas, como JavaScript<sup>1</sup>, HTML<sup>2</sup> e CSS<sup>3</sup>, além da utilização da plataforma do Google App Engine<sup>4</sup>, que vem crescendo e ganhando adeptos em todo o mundo, foram a motivação final para a escolha desse tema para a construção do Projeto Final de Conclusão de Curso.

---

<sup>1</sup> JavaScript - <http://www.w3schools.com/js>

<sup>2</sup> HTML - <http://www.w3schools.com/html>

<sup>3</sup> CSS - <http://www.w3schools.com/css>

<sup>4</sup> Google App Engine - <https://appengine.google.com>

## 1.2 - Objetivos

O cálculo de pontos de função hoje ainda é feito de forma predominantemente manual. As poucas ferramentas disponíveis para o cálculo são muito limitadas, visto que o conceito de ponto de função não é bem explorado por essas ferramentas.

O objetivo desse projeto é implementar uma ferramenta que seja capaz de fazer o cálculo de pontos de função a partir da análise de requisitos de um projeto. A ferramenta possibilitará que o usuário avalie o esforço que será necessário para a execução de um projeto, assim podendo tomar decisões estratégicas quanto a sua condução e viabilidade.

Além disso, a ferramenta possibilitará a exportação para um arquivo XML<sup>5</sup> de todas as informações sobre o projeto. Essa funcionalidade possibilitará a criação de futuras extensões para a ferramenta e a criação de novas funcionalidades.

## 1.3 - Organização do Texto

Além desta introdução, o trabalho será estruturado em capítulos e desenvolvido da seguinte forma:

- Capítulo II: Gerência de Projetos - apresentará os conceitos gerais sobre a disciplina de gerenciamento de projetos, apresentando as suas definições e das áreas de conhecimento do ponto de vista do PMBOK [PMBOK, 2000]. Após isso, focará na área de conhecimento que será tratada nesse trabalho, o gerenciamento de tempo, definindo e apresentando os conceitos referentes à Análise de Pontos de Função, tema principal do trabalho;
- Capítulo III: Tecnologias e Requisitos do Sistema – apresentará os conceitos básicos do ambiente que será utilizado para desenvolvimento e hospedagem do sistema, incluindo o seu modelo de persistência de dados e limitações. Também apresentará um detalhamento das funcionalidades que serão implementadas;

---

<sup>5</sup> XML - <http://www.w3schools.com/xml/>

- Capítulo IV: Sistema de Cálculo de Esforço para projetos de desenvolvimento de software baseado no método de Pontos de Função – apresentará o sistema desenvolvido com o objetivo de possibilitar o cálculo de esforço de projetos de desenvolvimento de software, baseado no método de Pontos de Função. Apresentará o fluxo de utilização da ferramenta, suas telas e funcionalidades implementadas;
- Capítulo V: Conclusões – apresentará as considerações finais do projeto, suas contribuições e algumas possibilidades de evolução futura.

# **2 Gerência de Projetos**

Com a popularização da prática de execução de projetos em diversas áreas de conhecimento, surgiu a necessidade de padronização e definição de boas práticas de vários conceitos em todas as áreas que envolvem o ciclo de vida de um projeto, criando a disciplina que hoje conhecemos como Gerenciamento de Projetos.

Esse capítulo descreverá os pontos básicos sobre Gerenciamento de Projetos. Iniciará definindo a base para o entendimento de o que é um projeto, mostrará as áreas de conhecimento envolvidas no gerenciamento de projeto de acordo com o PMBOK [PMI, 2012] e definirá os conceitos sobre a área de conhecimento que será abordada nesse projeto.

Por último, o capítulo apresentará os conceitos e detalhará a técnica de análise de Pontos de Função de acordo com o [IFPUG, 2010].

## **2.1 - Definição de Projeto**

Projeto é um empreendimento não repetitivo, caracterizado por uma sequência clara e lógica de eventos, com início, meio e fim, que se destina a atingir um objetivo claro e definido, sendo conduzido por pessoas dentro de parâmetros pré-definidos de tempo, custo, recursos envolvidos e qualidade [VARGAS, 2006].

Assim, um projeto é definido por um conjunto de ações e atividades que são executadas de maneira coordenada por uma “Organização Transitória”, na qual são alocados os insumos e recursos necessários para alcançar o objetivo determinado, em um prazo predefinido. A “Organização Transitória” é um conceito relacionado a um esquema particular e temporário que somente existe para tornar o trabalho com projetos mais eficiente e intuitivo por parte da organização [VARGAS, 2006].

## **2.2 - Áreas de Gerenciamento de Projeto**

As Áreas do Gerenciamento de Projetos ajudam a descrever os processos que compõem o gerenciamento como um todo. O Gerenciamento de projetos é dividido

em dez áreas de conhecimento. Cada área possui um escopo próprio e abrangência definida, sendo que a todo o momento está se integrando com as demais, formando um corpo de projeto organizado.

- Gerenciamento de Integração: Envolve a coordenação e integração de todos os processos que pertencem às outras áreas de conhecimento, garantindo que o projeto como um todo sempre seja beneficiado;
- Gerenciamento de Escopo: Define as limitações do projeto que está sendo planejado, determinando as funcionalidades que serão implantadas e as atividades necessárias para sua implantação, garantindo assim que no projeto esteja incluído todo o esforço necessário e somente o necessário para o sucesso do projeto;
- Gerenciamento de Tempo: Responsável por garantir o planejamento correto do tempo e esforço necessário para a conclusão do projeto no prazo definido;
- Gerenciamento de Custo: Engloba todos os processos para garantia de que o projeto seja executado e concluído de acordo com o orçamento determinado;
- Gerenciamento de Qualidade: Garante que os produtos e entregáveis do projeto estão de acordo com os requisitos solicitados pelo cliente;
- Gerenciamento de Recursos Humanos: Responsável por garantir a devida alocação e o uso dos profissionais envolvidos no projeto;
- Gerenciamento de Comunicação: Responsável por garantir que as informações necessárias para o correto andamento do projeto sejam distribuídas;
- Gerenciamento de Risco: Tem como objetivo o reconhecimento, qualificação, resposta e monitoramento dos riscos que podem gerar impacto para o sucesso do projeto;
- Gerenciamento de Aquisições: Responsável pela aquisição de bens e serviços externos necessários para o sucesso do projeto;
- Gerenciamento de Patrocinadores: Responsável por definir, planejar e controlar o engajamento das partes interessadas no projeto, garantindo o envolvimento e a comunicação dos interessados no projeto.

Esta monografia abordará um assunto relacionado a Gerenciamento de Tempo, sendo foco da mesma as técnicas de estimativa de esforço utilizando a Análise de Pontos de Função [IFPUG, 2010].

## **2.3 - Gerenciamento de Tempo**

O Gerenciamento de tempo inclui os processos necessários para garantir a realização completa do projeto no tempo que lhe foi definido. De acordo com o PMBOK [PMI, 2012], essa área de conhecimento inclui os seguintes processos:

- Definição de Atividades - Identifica as atividades que precisam ser feitas para produzir os resultados do projeto;
- Sequenciamento das Atividades – Determina a interdependência entre as atividades identificadas;
- Estimativa de Recursos para as Atividades – Define o tipo e a quantidade de recursos necessários para desempenhar cada atividade reconhecida;
- Estimativa de Duração para as Atividades - Determina o tempo necessário para o cumprimento de cada atividade, tendo em vista o conhecimento da equipe;
- Elaboração de Cronograma – Onde é feita a análise das durações, dependências, disponibilidade de recursos para cada atividade e, a partir disso, são determinadas as datas de início e fim para cada atividade mapeada;
- Controle de cronograma – Controla as mudanças no cronograma definido para o projeto.

Esta monografia irá abordar um tema referente ao processo de Estimativa de Duração, utilizando uma técnica de análise de pontos por função como foco do estudo e da implementação do sistema. Após o cálculo dos pontos de função para um determinado projeto, o resultado calculado servirá de insumo para a criação da estimativa de duração para as atividades definidas no projeto.

## **2.4 - Análise de Pontos por Função**

Esta seção descreve a base teórica sobre a técnica de contagem por pontos de função. Serão abordados temas como a necessidade de mecanismos de medição de esforço, o propósito desses mecanismos e os fundamentos para a contagem correta de Pontos por Função.

## 2.4.1 - A necessidade de medição

Nos dias atuais, um dos maiores problemas dos gestores de TI é a definição de indicadores abrangentes o suficiente para garantir o completo atendimento das demandas do usuário, garantindo assim a qualidade do produto. Sendo a demanda por produtos e serviços de TI cada vez maior, são necessários meios adequados de medição e a definição de métricas que devem ser seguidas. Essas métricas geram insumos para os gestores tomarem decisões mais seguras, diminuindo assim, o número de compromissos assumidos sem o conhecimento dos recursos que serão necessários para o atendimento.

Entre tantas métricas, a taxa de produtividade em TI é uma das mais importantes. Ela representa a razão entre a quantidade liberada de produtos e o esforço despendido na sua execução. Essa métrica ajuda a reconhecer oportunidades de melhoria, visto que representa a produtividade atual das unidades produtoras e pode ser comparada com a meta pretendida para assim serem traçados planos com o objetivo de aproveitar as oportunidades encontradas.

Em linhas tradicionais de produção, essa métrica é facilmente verificada, visto que é possível fazer a contagem de quantos produtos são gerados no final da linha de produção e as atividades de produção muitas vezes são padronizadas. Em TI, esse tipo de contagem é mais difícil, uma vez que as aplicações são bastante diferentes umas das outras e a velocidade e qualidade das tarefas executadas depende muito da capacidade dos profissionais envolvidos.

Dessa maneira, a produtividade de um departamento de desenvolvimento de sistemas precisa ser avaliada por uma técnica que seja independente da aplicação que é construída, da capacidade de cada profissional e das técnicas utilizadas.

A técnica de análise por Ponto de Função, desenvolvida por Allan Albrecht da IBM, auxilia os gerentes TI a medir a produtividade relacionada ao esforço realizado no processo de desenvolvimento ou manutenção de sistemas. [DIAS, RAQUEL, 2003]. Utilizando essa técnica, podemos estimar o esforço despendido por atividade ou unidade em um departamento de TI. Além disso, podemos obter insumos para a compreensão do comportamento verificado em projetos concluídos ou em andamento,

entendendo melhor as falhas e problemas, para assim podermos planejar projetos futuros de forma mais eficiente.

### **2.4.2 - Objetivos da contagem por Pontos de Função**

Todo sistema tem como objetivo final o atendimento de um conjunto de requisitos de usuário, executando ações e processos definidos pelos mesmos para que algum objetivo específico seja alcançado. Isso independe do tipo de sistema, do ambiente em que foi construído e das técnicas utilizadas para construí-lo. A técnica de contagem por pontos de função produz uma métrica de produtividade baseada na visão de funcionalidades do usuário.

Sendo assim, o dimensionamento do sistema é considerado sob o ponto de vista do usuário, não depende da tecnologia empregada para o desenvolvimento ou do conhecimento, habilidade e experiência do grupo que desenvolveu ou fez a manutenção do sistema. [DIAS, RAQUEL, 2003].

Sendo a métrica independente de questões técnicas relacionadas com a gestão de TI, torna possível comparações entre sistemas, combinando os Pontos por Função com outras métricas que podem gerar diversas informações sobre a produtividade das unidades, como: custo por produtividade, qualidade por produtividade, e assim, ajudar na elaboração de planos de ação para o atendimento das metas de produtividade estabelecidas.

### **2.4.3 - Conceitos de Pontos de Função**

Conforme exposto anteriormente, Pontos por Função é uma métrica baseada na visão das funcionalidades do usuário. Suas principais características são:

- Independe das tecnologias e técnicas utilizadas para a produção;
- Foco nos requisitos do sistema e funcionalidades na visão do usuário;
- É um processo de contagem que possibilita a automação.

Os benefícios pelo uso da Análise de Pontos de Função são:

- Possibilita o dimensionamento do sistema;

- Apóia nas estimativas de custos e recursos necessários para manutenção e construção de um sistema;
- Apóia na tomada de decisão para aquisição de serviços e produtos.

A principal desvantagem dessa métrica é que os dados gerados possuem uma dose de subjetividade, visto que representam a visão do usuário e não podem ser representados fisicamente, necessitando de uma análise comparativa com outros projetos para que sejam geradas métricas relevantes para o cliente.

A complexidade do sistema sugerida por essa técnica está relacionada com os cinco fatores listados abaixo:

- A quantidade de arquivos lógicos;
- A quantidade de registros lógicos;
- A quantidade de itens de dados identificados;
- A quantidade de transações previstas para o sistema;
- A quantidade de dados e registros manipulados por essas transações.

#### **2.4.4 - Etapas do processo de avaliação**

O processo de avaliação de um projeto e contagem de Pontos de Função é dividido em 3 (três) etapas, sendo elas:

- Identificar que tipo de contagem será utilizada: Essa etapa é composta pela identificação do que será medido. Existem três tipos de contagem: contagem de PF de projeto de desenvolvimento, de aplicações instaladas e de projetos de manutenção [HAZAN, 2001];
- Definir as fronteiras da aplicação: Nessa etapa é definido o limite do sistema, ou seja, quais funcionalidades do ponto de vista do usuário serão atendidas dentro do projeto e devem ser consideradas na avaliação por Pontos de Função. Somado a essa avaliação, também são definidos os relacionamentos entre o sistema e outras aplicações externas;
- Contar Pontos de Função: A partir das análises feitas anteriormente será iniciado o processo de contagem. Nesse processo, serão identificadas as funções que serão implementadas para compor o sistema. Essas funções representam os arquivos

acessados pelo sistema e as transações que irão manipular esses dados. Essas funções participaram no processo de contagem do número de Pontos de Função calculado para o projeto.

Sendo assim, a contagem que gera um número de Pontos de Função que representará o esforço calculado para a implementação dos requisitos solicitados pelo usuário. As funções disponibilizadas para o usuário são definidas como:

- ILF – *Internal Logical File* (Arquivo Lógico Interno) – dados que serão utilizados pela aplicação e são mantidos por ela;
- EIF – *External Interface File* (Arquivo de Interface Externa) – dados que são utilizados pela aplicação, porém não são mantidos por ela;
- EI – *External Input* (Entrada Externa) – transações que controlam dados que são originados de fora da aplicação;
- EO – *External Output* (Saída Externa) – envia dados processados para fora da aplicação;
- EQ – *External Query* (Consulta Externa) – envia dados dos ILFs ou EIFs para fora da aplicação sem nenhum processamento.

Após a identificação do tipo de contagem que será feita, devemos prosseguir definindo os subgrupos de Funções de Dados e Funções Transacionais.

- Funções de Dados – São todas as funções classificadas com ILF ou EIF. É a funcionalidade fornecida ao usuário para atender às necessidades de armazenamento de dados internos ou externos [IFPUG, 2010];
- Funções Transacionais – São todas as funções classificadas como EI, EO ou EQ – processo elementar que provê ao usuário funcionalidades de processamento de dados [IFPUG, 2010].

A partir desse momento faremos a contagem de Pontos de Função para cada função identificada. Para cada função devemos contar os seguintes itens:

- RET (*Record Element Type*) – é um subgrupo de elementos de dado reconhecíveis pelo usuário dentro de uma Função de Dados [IFPUG, 2010];

- FTR (*File Type Referenced*) – é uma Função de Dados lida ou mantida por uma Função Transacional [IFPUG, 2010];
- DET (*Data Element Type*) – é um atributo único, reconhecível pelo usuário e não repetitivo [IFPUG, 2010].

### 2.4.5 - Contagem de Pontos de Função de Dados

Um ILF é um conjunto de dados ou informações de controle escolhidos pelo usuário da aplicação como necessidades de informação. Sua manutenção é feita dentro dos limites da aplicação e por processos elementares da mesma (Inclusão, Alteração, Remoção). Seu objetivo é manter as informações que serão utilizadas pela aplicação.

Um DET deve ser contado para cada campo reconhecido pelo usuário, de acordo com as regras: (i) campos em duplicada são contados uma única vez; (ii) chaves primárias que não representem informações percebidas pelo usuário não são contadas como DET; (iii) chaves estrangeiras que servem para relacionar um elemento com outro ILF/EIF devem ser contadas com um DET; (iv) o resultado de um cálculo conta com um DET; e (v) em um EIF, devem ser contados apenas os campos manipulados pela aplicação.

A contagem de RETs deve ser feita para cada subgrupo de elementos de dados dentro de um ILF ou EIF. Deve ser contado um RET para cada subgrupo opcional ou mandatório de um ILF ou EIF. Caso esses subgrupos não existam, só deve ser contado um único RET representando o próprio ILF ou EIF.

A complexidade de um ILF é determinada a partir da quantidade de elementos de dados (DET) e registros lógicos (RET) referenciados dentro da função. De posse destas informações, utilize a Tabela 1 para definir a complexidade do ILF analisado, de acordo com seu número de DETs e RETs, e a Tabela 2 para encontrar a quantidade de pontos de função atribuída a esse ILF.

RET/DET	1/19 DET	20/50 DET	51+ DET
---------	----------	-----------	---------

1 RET	Baixa	Baixa	Média
2/5 RET	Baixa	Média	Alta
6+ RET	Média	Alta	Alta

Tabela 1 – Nível de Complexidade de um ILF

Complexidade	ILF PF
Baixa	7
Média	10
Alta	15

Tabela 2 – Contribuição de cada ILF para a contagem de Pontos de Função

Um EIF é um conjunto de dados necessários para a aplicação, mantidos fora dos limites da aplicação em questão. Sua manutenção é feita fora dos limites da aplicação e seu objetivo principal é manter as informações utilizadas pela aplicação.

Assim como os ILFs, a complexidade de um EIF é determinada a partir da quantidade de elementos de dados (DET) e registros lógicos (RET) referenciados dentro da função. As regras de contagem são as mesmas aplicáveis aos ILF. De posse destas informações, utilize a Tabela 3 para definir a complexidade do EIF analisado e após isso a Tabela 4 para encontrar a quantidade de pontos de função atribuída a esse EIF.

RET/DET	1/19 DET	20/50 DET	51+ DET
1 RET	Baixa	Baixa	Média
2/5 RET	Baixa	Média	Alta
6+ RET	Média	Alta	Alta

Tabela 3 – Nível de Complexidade de um EIF

Complexidade	EIF PF
Baixa	5
Média	7
Alta	10

Tabela 4 – Contribuição de cada EIF para a contagem de Pontos de Função

## 2.4.6 - Contagem de Pontos de Função Transacionais

Um EI é um conjunto de dados que são inseridos na aplicação a fim de manter um dos ILFs da aplicação. Sendo assim, eles provocam a inclusão, alteração ou remoção dos dados de um ILF. São identificados a partir de processos que recebem dados externos.

A complexidade de um EI é determinada a partir da quantidade de elementos de dados (DET) e Arquivos Internos ou Interfaces Externas (FTR) referenciados dentro do processo. Para cada EI deve ser contado um FTR para cada Arquivo Interno mantido, um FTR para cada Arquivo Interno lido, um FTR para cada Interface Externa lida e apenas um FRT para Arquivos ou Interfaces que são lidas e mantidas. Em relação aos DET, deve ser contado um DET para cada campo reconhecível pelo usuário e único, um DET para todas as mensagens de erro e confirmação que podem ser emitidas pelo processo e um DET para todas as formas que o processo pode ser disparado pelo usuário, quando houver mais de uma forma.

De posse das informações sobre DET e FTR, utilize a Tabela 5 para definir a complexidade do EI analisado e após isso a Tabela 6 para encontrar a quantidade de pontos de função atribuída a esse EI.

EI	1/4 DET	5/15 DET	16+ DET
0/1 FTR	Baixa	Baixa	Média
2 FTR	Baixa	Média	Alta
6+ FTR	Média	Alta	Alta

**Tabela 5 – Nível de complexidade de um EI**

Complexidade	EI PF
Baixa	3
Média	4
Alta	6

**Tabela 6 – Contribuição de cada EI para a contagem de Pontos de Função**

Um EO é um processo da aplicação que usa dados dos Arquivos Internos e Interfaces Externas, os transforma e gera saídas de interesse do usuário. O objetivo principal é apresentar informações que foram processadas para o usuário.

A complexidade de um EO é determinada a partir da quantidade de elementos de dados (DET) e Arquivos Internos ou Interfaces Externas (FTR) referenciados dentro do processo. A contagem destas informações segue o mesmo modelo usado nos EI. Utilize a Tabela 7 para definir a complexidade do EO analisado e após isso a Tabela 8 para encontrar a quantidade de pontos de função atribuída a esse EO.

EO	1/5 DET	6/19 DET	20+ DET
0/1 FTR	Baixa	Baixa	Média
2/3 FTR	Baixa	Média	Alta
4+ FTR	Média	Alta	Alta

**Tabela 7 – Nível de complexidade de um EO**

Complexidade	EO PF
Baixa	4
Média	5
Alta	7

**Tabela 8 – Contribuição de cada EO para a contagem de Pontos de Função**

Por fim, um EQ é um processo da aplicação que recebe parâmetros e recupera informações solicitada pelo usuário. Essa informação não é alterada, sendo mostrada para o usuário exatamente como está armazenada e sem nenhum tipo de alteração nos arquivos internos. A complexidade de um EQ é determinada a partir da quantidade de elementos de dados (DET) e Arquivos Internos ou Interfaces Externas (FTR) referenciados dentro do processo. Utilize a Tabela 9 para definir a complexidade do EQ analisado e após isso a Tabela 10 para encontrar a quantidade de pontos de função atribuída a esse EQ.

EQ	1/5 DET	6/19 DET	20+ DET
0/1 FTR	Baixa	Baixa	Média
2/3 FTR	Baixa	Média	Alta
4+ FTR	Média	Alta	Alta

**Tabela 9 – Nível de Complexidade de um EQ**

Complexidade	EQ PF
Baixa	3
Média	4
Alta	6

**Tabela 10 – Contribuição de cada EQ para a contagem de Pontos de Função**

Após a contagem de pontos de função para todas as 5 categorias de funções apresentadas, a contribuição de cada uma delas deve ser somada, gerando assim o total de pontos de função do projeto.

## **2.4.7- Utilização dos Pontos de Função para Geração de Métricas**

A partir do cálculo de Pontos de Função, diversas métricas poderão ser elaboradas, o utilizando como base. Diversas características do projeto podem ter como base a Análise de Pontos de Função. Abaixo, citamos algumas destas características.

- $\text{Número de Horas Total do Projeto} / \text{Pontos de Função} = \text{Produtividade}$
- $\text{Tempo de Calendário de Projeto} / \text{Pontos de Função} = \text{Taxa de Entrega}$
- $\text{Custo Total do Projeto} / \text{Pontos de Função} = \text{Custo}$
- $\text{Número de Defeitos} / \text{Pontos de Função} = \text{Qualidade}$
- $\text{Número de Páginas de Documento} / \text{Pontos de Função} = \text{Documentação}$
- $\text{Número de Casos de Teste} / \text{Pontos de Função} = \text{Testes}$ .

Sendo assim, Pontos de Função pode ser uma medida base para diversas características de um projeto ou de uma unidade de negócio que ajudará na definição de metas e no acompanhamento de indicadores.

## **2.5 – Considerações Finais**

Esse capítulo apresentou os conceitos de gerenciamento de projeto, focando principalmente na área de gerenciamento de tempo. Apresentou também a Análise de Pontos de Função, as etapas de análise e as técnicas utilizadas para essa análise.

O próximo capítulo apresentará as tecnologias utilizadas no desenvolvimento do sistema tratado nesse projeto, além de uma descrição dos requisitos que descrevem o escopo do sistema.

# 3 Tecnologias e Requisitos do Sistema

Nesse capítulo serão apresentados os requisitos do sistema implementado durante o projeto, descrevendo seus casos de uso, formas de utilização e um modelo conceitual desse. Além disso, serão descritas as tecnologias utilizadas para a construção do sistema, sendo elas o Google App Engine<sup>6</sup> e um banco de dados NoSQL<sup>7</sup>. Serão apresentados os conceitos, vantagens e desvantagens e as principais características dessas tecnologias. Por fim, serão descritos e apresentados o ambiente de desenvolvimento utilizado para a implementação do projeto e a linguagem utilizada durante o desenvolvimento.

## 3.1 - Requisitos do Sistema

O sistema construído durante o desenvolvimento desse projeto prevê atender aos requisitos presentes no diagrama de casos de uso apresentado na Figura 1

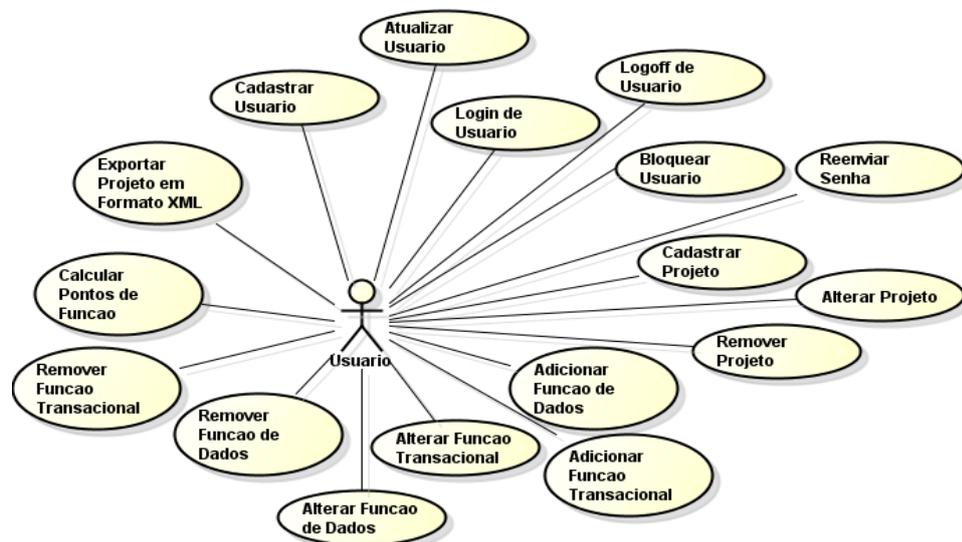


Figura 1 – Diagrama de Casos de Uso do Sistema

A seguir, a descrição dos requisitos contemplados pelo sistema:

<sup>6</sup> GAE - <https://developers.google.com/appengine/docs/whatisgoogleappengine?hl=pt-br>

<sup>7</sup> NoSQL - <http://nosql-database.org/>

- Cadastrar Usuário – Para utilização do sistema é necessário estar cadastrado como um dos usuários do mesmo. Esse caso de uso permitirá o cadastro de um novo usuário no sistema;
- Atualizar Usuário – Esse caso de uso permite a atualização das informações cadastrais de um usuário já existente no sistema, permitindo que seus dados pessoais sejam corrigidos em caso de mudança;
- Login do Usuário – Esse caso de uso permite que um usuário já cadastrado no sistema acesse suas funcionalidades, a partir da comparação dos dados fornecidos para acesso com os dados cadastrados no banco de dados do sistema. Assim, o usuário poderá fazer o cadastro de projetos e o cálculo de Pontos de Função;
- Logoff de Usuário – Esse caso de uso permite que um usuário logado no sistema, faça o logoff e fique fora do ambiente do sistema;
- Bloquear Usuário – Esse caso de uso permite que um usuário seja bloqueado do sistema, caso tenha tentando logar no sistema três vezes sem sucesso;
- Reenviar Senha – Esse caso de uso permite que o usuário tenha a sua senha reenviada para o seu e-mail caso ele tenha esquecido a mesma. O reenvio ocorre para o e-mail cadastrado no sistema;
- Cadastrar Projeto – Esse caso de uso permite que um usuário logado no sistema faça o cadastro de um novo projeto. A partir desse cadastro poderão ser incluídas as informações desse projeto, como funções de dados e transacionais, que servirão de insumo para o cálculo de Pontos de Função do projeto;
- Alterar Projeto – Esse caso de uso permite que sejam alteradas as informações de um projeto já cadastrado no sistema, assim sendo possível a alteração do nome cadastrado para o projeto;
- Remover Projeto – Esse caso de uso permite que um projeto já cadastrado no sistema seja excluído de seu banco de dados. Com o projeto excluído, todas as informações de funções de dados ou transacionais relacionadas ao mesmo também serão excluídas;
- Adicionar Função de Dados – Esse caso de uso permite que seja criada uma função de dados relacionada a um projeto já existente. As informações dessa função de dados farão parte do cálculo de Ponto por Função do projeto;

- Adicionar Função Transacional – Esse caso de uso permite que seja criada uma função transacional relacionada a um projeto já existente. As informações dessa função transacional farão parte do cálculo de Ponto por Função do projeto;
- Alterar Função de Dados – Esse caso de uso permite que as informações de uma função de dados já cadastrada em um projeto sejam alteradas;
- Alterar Função Transacional – Esse caso de uso permite que as informações de uma função transacional já cadastrada em um projeto sejam alteradas;
- Remover Função de Dados – Esse caso de uso permite que uma função de dados cadastrada em um projeto seja excluída e assim não faça mais parte do cálculo de Pontos de Função;
- Remover Função Transacional – Esse caso de uso permite que uma função transacional cadastrada em um projeto seja excluída e assim não faça mais parte do cálculo de Pontos de Função.
- Calcular Pontos de Função – Esse caso de uso permite que seja feito o cálculo dos Pontos de Função de um projeto a partir de suas funções de dados e transacionais já cadastradas;
- Exportar Projeto em Formato XML – Esse caso de uso permite a exportação em formato XML de todas as informações de um projeto já cadastrado, incluindo suas funções de dados e transacionais.

## **3.2 - Modelo Conceitual**

Após a análise dos requisitos e antes do início do desenvolvimento foi feito um modelo conceitual do sistema implementado durante o projeto. A Figura 2 apresenta as classes do domínio.

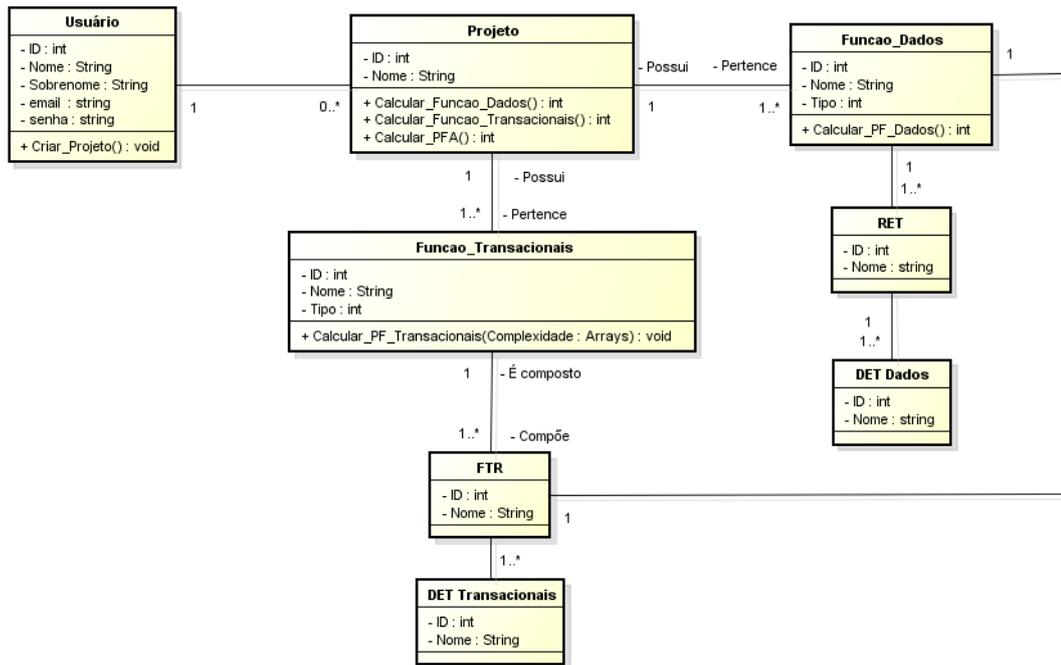


Figura 2 – Diagrama de Classes do Sistema

A classe Usuario armazena as informações de um usuário do sistema. Esse usuário pode possuir diversos projetos cadastrados. A classe Projeto armazena os dados de um projeto, que pode possuir diversas Funções de Dados, representadas pela classe Funcao\_Dados, e diversas Funções Transacionais, representada pela classe Funcao\_Transacional. As funções de dados podem possuir diversos RETs, que por sua vez podem possuir diversos DETs, representados pelas classes RET e DET\_Dados, respectivamente. As funções transacionais podem possuir diversos FTRs, que por sua vez podem possuir diversos DETs, representados pelas classes FTR e DET\_Transacionais, respectivamente. Além disso, um FTR é a representação de um função de dados do sistema e, por esse motivo, possui um relacionamento com uma função de dados.

### 3.3 - Google App Engine

Google App Engine (ou GAE) é uma plataforma de desenvolvimento e hospedagem de aplicações web baseada nos conceitos de plataforma como serviço<sup>8</sup>

<sup>8</sup> Plataforma como serviço - [http://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Platform_as_a_service)

(PaaS) e computação em nuvem, onde a hospedagem da aplicação é feita dentro dos data centers geridos pela Google.

**Plataforma como serviço (PaaS)** é uma capacidade provida ao consumidor de implantar em uma infraestrutura de nuvem aplicações criadas ou adquiridas pelo consumidor usando linguagens, bibliotecas, serviços e ferramentas suportadas pelo provedor. O consumidor não gerencia ou controla a infraestrutura da nuvem incluindo rede, servidores, sistemas operacionais ou armazenamento, mas possui o controle sobre a aplicação implantada e configurações possíveis no ambiente disponibilizado pelo servidor de aplicação [MELL e GRANCE, 2011].

As aplicações criadas para utilizar a plataforma da Google rodam sobre uma infraestrutura de nuvem composta por múltiplos servidores. O GAE oferece escalabilidade para as aplicações web, alocando automaticamente mais recursos conforme as requisições para as mesmas forem aumentando. Seu uso é gratuito até 500 MB de armazenamento e cerca de 5 milhões de requisições por mês. A partir desse limite taxas são cobradas para a utilização dos recursos adicionais necessários. O GAE oferece os seguintes recursos:

- Armazenamento persistente, possibilitando consultas, transações e classificação;
- Balanceamento de carga e escalonamento automático;
- APIs para autenticação e envio de e-mails através de contas Google;
- Ambiente de desenvolvimento local, com simulação de todos os recursos disponíveis no GAE;
- Tarefas agendadas para disparar eventos em horários específicos e com intervalos regulares.

As aplicações web são executadas em ambientes seguros, independentes do sistema operacional, hardware e localização física dos servidores web. Assim, o GAE pode distribuir as requisições entre os diversos servidores, aumentando ou diminuindo o número de servidores para atender o tráfego solicitado. Por outro lado, o serviço não é livre de restrições, como as apresentadas a seguir:

- O aplicativo somente pode acessar computadores na Internet através de e-mails ou serviços de busca por URL. Outros computadores só podem se comunicar com o aplicativos rodando no GAE através de requisições HTTP;
- O aplicativo somente pode ler arquivos armazenados junto ao seu código, não podendo gravar arquivos no sistema de arquivos. Sendo assim, o aplicativo deve utilizar o armazenamento de dados e outros serviços do GAE para persistência de dados entre as requisições.

A persistência de dados no GAE é feita através de um serviço de armazenamento de dados distribuído, que possibilita consultas e transações. Esse armazenamento cresce à medida que os dados aumentam. O armazenamento no GAE não é feito através de um banco de dados relacional, mas os objetos de dados são armazenados como entidades que possuem um tipo e um conjunto de propriedades. Sendo assim, uma consulta pode recuperar um conjunto de entidades de um determinado tipo através de filtros e classificadores de suas propriedades. O armazenamento é feito de forma consistente e utilizando o controle de concorrência otimista<sup>9</sup>. A aplicação pode executar diversas operações de armazenamento em uma única transação, sendo que todas terão sucesso ou falharão, assegurando a integridade dos dados.

O GAE oferece suporte para as linguagens Java<sup>10</sup>, Python<sup>11</sup>, PHP<sup>12</sup> e GO<sup>13</sup> e a Google possui planos para aumentar o número de linguagens suportadas no futuro.

O GAE foi utilizado como plataforma para a aplicação web desenvolvida durante o projeto e se apresentou como facilitador durante o processo. O autor desse projeto nunca havia passado pela experiência de desenvolvedor e o GAE se mostrou de grande ajuda por não exigir nenhum tipo de configuração ou preocupação com a infraestrutura que seria usada para rodar a aplicação. Dessa forma, a atenção do autor ficou totalmente voltada para o desenvolvimento da aplicação.

---

<sup>9</sup> Controle de concorrência otimista - [http://en.wikipedia.org/wiki/Optimistic\\_concurrency\\_control](http://en.wikipedia.org/wiki/Optimistic_concurrency_control)

<sup>10</sup> Java - [http://www.java.com/pt\\_BR/](http://www.java.com/pt_BR/)

<sup>11</sup> Python - <http://www.python.org.br/wiki>

<sup>12</sup> PHP - <http://php.net/>

<sup>13</sup> GO - [http://en.wikipedia.org/wiki/Go\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Go_(programming_language))

Além disso, o serviço de armazenamento disponibilizado pelo GAE é simples e de fácil utilização. Novamente, o autor desse projeto não possuía experiência em banco de dados NoSQL e sempre utilizou e administrou bancos de dados relacionais, como PostgreSQL<sup>14</sup> e MySQL<sup>15</sup>. Utilizando o GAE, o desenvolvedor não precisa atentar a questões comuns nos bancos de dados relacionais, como criação do banco, tabelas, esquemas e instanciação do banco de dados: deve pensar apenas na criação das entidades que vão representar os dados manipulados pela aplicação e suas propriedades. Somado a isso, as consultas são criadas sem a utilização do SQL e sim com a aplicação de filtros pelas propriedades de cada entidade, facilitando a construção de consultas dentro do base de dados da aplicação.

Outro recurso utilizado para o desenvolvimento da aplicação foi o serviço de e-mail disponibilizado pelo GAE. Esse serviço, possibilita que o desenvolvedor implemente funcionalidades e eventos dentro da aplicação de disparem e-mails para os seus usuários. Ele compõe o e-mail preenchendo os campos *sender* (remetente), *to* (destinatário), *subject* (assunto) e *body* (corpo) com os atributos referentes a cada um dos campos. Após a composição do e-mail o mesmo é enviado ao destinatário definido pelos servidores da Google. Para a utilização desse serviço, o e-mail utilizado como remetente precisa estar cadastrado como administrador da aplicação e o e-mail do destinatário precisa ter um formato válido.

### 3.4 - NoSQL

NoSQL é um termo utilizado para definir bancos de dados não-relacionais, que atualmente estão em grande utilização e desenvolvimento [ROCHA, 2010]. Esses bancos foram idealizados para suprir as demandas onde os bancos relacionais se mostravam ineficazes, como necessidades de alta escalabilidade e alta disponibilidade, visto que bancos NoSQL têm sua base de dados distribuídas entre vários servidores, o que evita a criação de um ponto de falha centralizado permitindo que, caso um servidor caia, o serviço não seja afetado e os usuários sejam direcionados para outros servidores.

---

<sup>14</sup> PostgreSQL - <http://www.postgresql.org.br/>

<sup>15</sup> MySQL - <http://www.mysql.com/>

Muitos deles apresentam outras características interessantes como replicação, escalabilidade horizontal, entre outras.

Os bancos de dados relacionais possuem características ainda muito restritas, pois utilizam o escalonamento vertical dos servidores, ou seja, quanto mais dados, mais espaço no servidor e memória são necessários [ROCHA, 2010], o que torna muito complexo o processo de torna-lo altamente escalável.

O NoSQL provê maior escalabilidade, pois utiliza distribuição horizontal, ou seja, quanto mais dados mais servidores serão alocados para suporta-los, mas não necessariamente servidores de alto desempenho. Essa forma de utilização é muito mais eficiente e econômica. Outra vantagem, é que devido a divisão dos dados em vários servidores, o volume de dados por servidor é minimizado, tornando mais fácil o armazenamento, gerenciamento e processamento desses dados.

Os bancos NoSQL não exigem esquemas de tabela e não armazenam seus dados de maneira relacional, onde várias tabelas são necessárias para armazenar uma informação e essa é recuperada através de consultas explorando os relacionamentos entre essas tabelas. Sendo assim, uma característica dos bancos NoSQL é a falta de suporte a operação de join do SQL. Em casos em que o join é necessário, o NoSQL propõe o armazenamento de toda a informação necessário no mesmo registro, ainda que isto exija replicação dos dados.

Entre as principais características de bancos de dados NoSQL, podemos citar:

- Escalabilidade horizontal: permite adicionar mais nós ao sistema, ou seja, mais servidores a um sistema de software que permita melhor distribuição do trabalho entre as máquinas;
- Replicação: permite a cópia de informações em mais de um nó para aumentar a capacidade de recuperação das informações;
- Schema-free: um dos fatores que contribuem para melhor escalabilidade dos bancos NoSQL é a ausência de Schemas. A vantagem principal desse conceito é permitir que as aplicações façam upgrades rápidos na estrutura dos dados sem ter que executar custosas reescritas e atualizações de estrutura em tabelas. Também

proporciona grande flexibilidade em armazenar dados estruturados de forma heterogênea [ZYP, 2010].

- Clusterização: o banco de dados é armazenado e gerenciado por mais de um servidor, aumentando a disponibilidade e desempenho do sistema;
- Sharding: consiste em dividir os dados horizontalmente, quebrando as tabelas e diminuindo o número de linhas armazenadas em um mesmo nó, separando estas linhas em ambientes diferentes.

Esse tipo de banco de dados é muito utilizado em serviços que dependem de armazenamento em nuvem, visto que ele não exige a alocação de um único servidor para persistência dos dados. Assim, ele vem se popularizando com o aumento de aplicações que utilizam de processamento de quantidades massivas de dados e por prover escalabilidade e desempenho maiores que os bancos relacionais. Os bancos de dados NoSQL podem ser divididos nas seguintes classificações [ROCHA, 2010]:

- Baseados em chave-valor: um coleção de chaves únicas e valores que são associados a essas chaves;
- Orientados a documentos: utiliza documentos como unidade básica de armazenamento. Os documentos não possuem qualquer tipo de estrutura pré-definida;
- Orientados a colunas: muda-se a orientação dos registros para colunas, ou seja, diferente dos bancos de dados tradicionais (orientados a linhas), onde os dados de um registro são armazenados em uma única linha em uma tabela, os bancos orientados a coluna são caracterizados assim pois cada valor de uma coluna (ou atributo) é armazenado continuamente [NORWAY, 2005]. A Figura 3, abaixo, apresenta uma comparação entre os modelos de armazenamento em linha e em coluna;



Figura 3 – Comparação entre Banco Orientado a Linhas e Orientado a Colunas

- Baseados em grafos: dados são armazenados em nós de grafos, onde as arestas representam o tipo de associação entre os nós.

Um grande utilizador do conceito de banco de dados NoSQL é o Google, que investe desde 2004 no banco de dados BigTable, desenvolvido pelo próprio Google para suprir as necessidades de armazenamento massivo da empresa e é totalmente baseado na filosofia de alto desempenho, escalabilidade e disponibilidade. O Google utiliza computadores de pequeno e médio, portes distribuídos em sua rede para armazenamento de seus dados.

### 3.5 - Ambiente de desenvolvimento e linguagem

A linguagem de programação utilizada no desenvolvimento da aplicação foi o Java, criada no ano de 1995 por James Gosling da Sun Microsystem, hoje propriedade da Oracle [ORACLE, 2011]. Java é uma linguagem orientada a objetos, derivada das linguagens C e C++ e apresenta algumas características interessantes que motivam o seu uso, como: operações de baixo nível automatizadas, um exemplo é a liberação de memória que é feita de forma automática através da de uma rotina nativa do java chamada *garbage collector*; facilidade de internacionalização; criação automática de ponteiros; e um vasto conjunto de bibliotecas para as mais diversas necessidades. Diferente de outras linguagens, que quando compiladas se tornam código nativo, a linguagem Java é compilada para um *bytecode* que é executado em uma máquina virtual, permitindo que após compilado, o sistema possa ser executado em qualquer computador, independente do sistema operacional e arquitetura.

O ambiente de desenvolvimento utilizado durante o projeto foi o Eclipse<sup>16</sup>, uma IDE (*Integrated development enviroment*) multi-linguagem e com um vasto conjunto de plug-ins. Ele foi originalmente projetado para o desenvolvimento em Java, mas através da utilização dos plug-ins pode ser utilizado para o desenvolvimento em diversas linguagens. Alguns plug-ins também podem ser utilizados para agregar outras funcionalidades ao Eclipse, como integração com outros sistemas e modelagem. Além disso, o Eclipse é um ambiente de desenvolvimento multi-plataforma, visto que pode

---

<sup>16</sup> Eclipse - <http://www.eclipse.org/>

ser utilizado em diversos sistemas operacionais, como Windows, Linux e OSX. O Eclipse é um software gratuito e com o código aberto.

Durante o projeto foi utilizado um plug-in desenvolvido pelo Google para o Eclipse. O Google Plug-in possibilita que os desenvolvedores criem e implantem aplicações utilizando o GAE [GOOGLE, 2013]. O Google Plug-in é um conjunto de ferramentas de desenvolvimento que possibilita aos desenvolvedores Java que modelem, construam, otimizem e implementem aplicações baseadas em nuvem [GOOGLE, 2013]. Esse plug-in possui diversas bibliotecas que são de extrema ajuda ao desenvolvedor que queria utilizar o GAE como plataforma web, como bibliotecas de acesso ao banco de dados, de autenticação, controle de sessão, entre outras. Além disso, o Google Plug-in permite que o desenvolvedor rode a aplicação web em ambiente de desenvolvimento local, simulando o ambiente do GAE na máquina do desenvolvedor. Por fim, o Google Plug-in possui uma ferramenta de implantação que permite o versionamento da aplicação web.

Durante o desenvolvimento o autor desse projeto teve algumas dificuldades ao configurar o Eclipse para rodar o Google Plug-in, pois os erros apresentados pelo compilador ainda são pouco explicativos, exigindo algum esforço em pesquisas para que sejam encontradas soluções para alguns erros apresentados durante a implantação da aplicação web no ambiente do GAE. Um exemplo dessas dificuldades foi a necessidade de exclusão de todas as JREs presentes na máquina de desenvolvimento, pois o Google Plug-in apenas funciona com JDKs instaladas. Porém, resolvidas estas dificuldades iniciais, o desenvolvimento seguiu sem grandes contratempos.

### **3.6 - Considerações Finais**

Esse capítulo tratou do desenvolvimento do sistema de cálculo de pontos de função, passando pelo levantamento de seus requisitos, modelagem do sistema e a descrição das tecnologias utilizadas durante o projeto, mostrando como essas tecnologias se mostraram úteis durante todo o desenvolvimento da aplicação. O próximo capítulo apresentará o fluxo de utilização do sistema, com suas principais funcionalidades e telas.

# 4 Sistema de cálculo de esforço para projetos de software baseado em Pontos de Função

O capítulo 4 apresentará o fluxo de utilização da ferramenta implementada durante o projeto apresentando suas principais telas e suas características.

## 4.1 - Cadastro de Usuário

Ao acessar o sistema pela primeira vez o usuário será encaminhado para a tela de login da ferramenta. A Figura 4 apresenta essa tela.

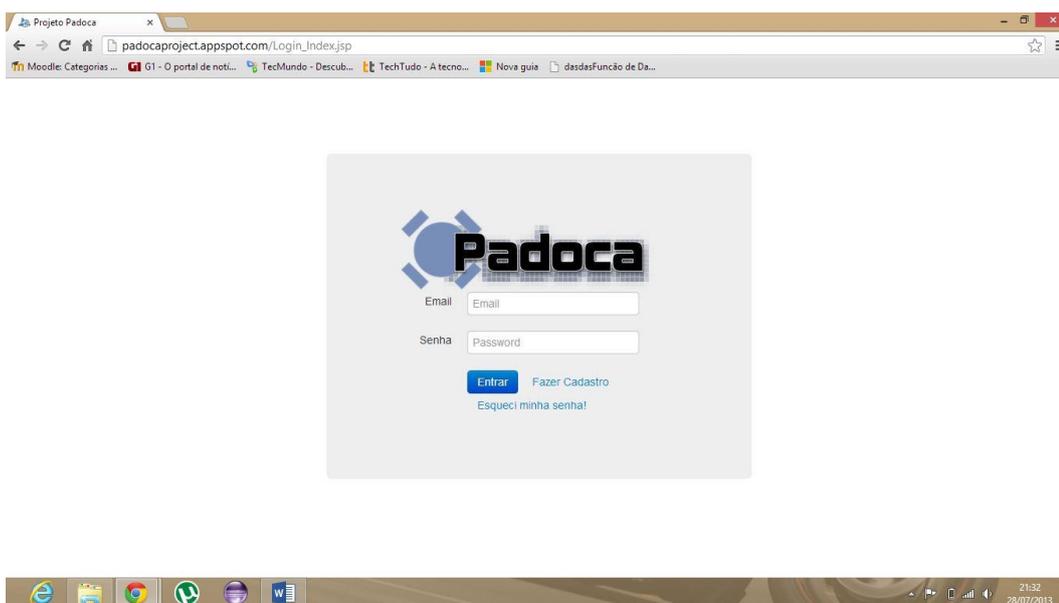
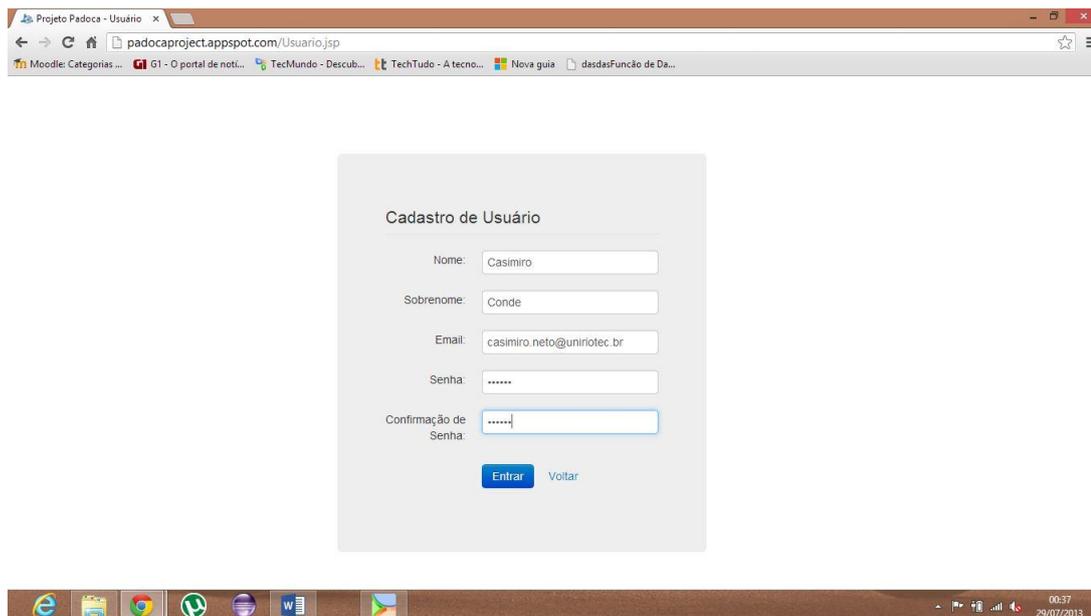


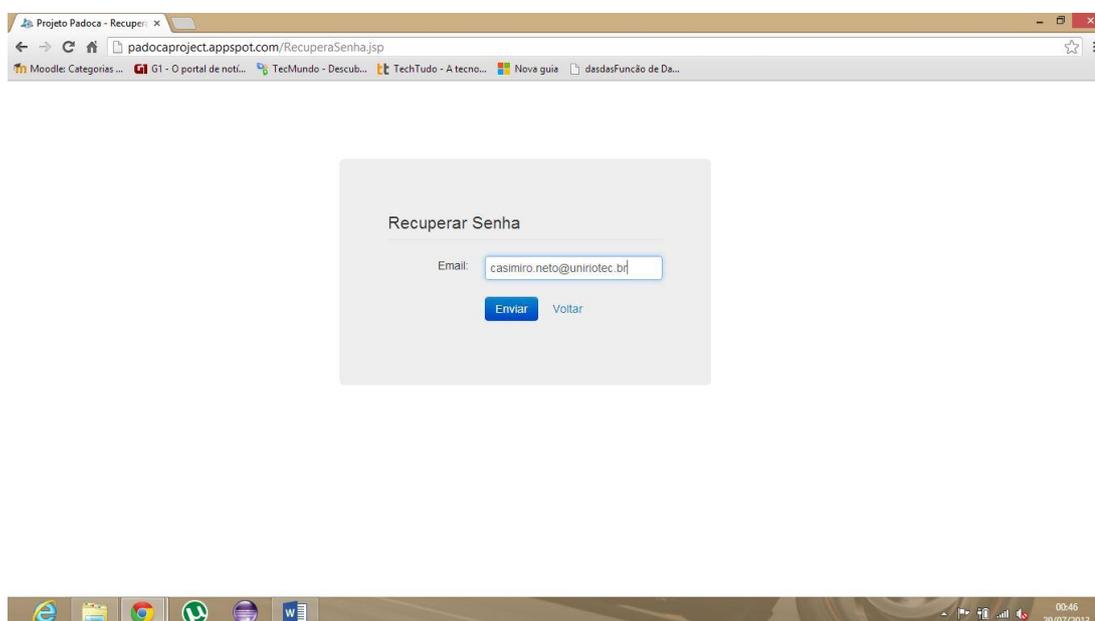
Figura 4 – Tela de Login

Na tela de login, o usuário já cadastrado poderá fazer entrar sua identificação e senha para acessar o sistema. Caso o usuário não possua cadastro no sistema poderá utilizar a opção “Fazer Cadastro”, que o levará para a tela de cadastro de usuário. Nesta tela, ele preencherá o formulário de cadastro com suas informações e informará o e-mail para login e sua senha de acesso. A tela de cadastro de usuário é apresentada pela Figura 5.



**Figura 5 – Tela de Cadastro de Usuário**

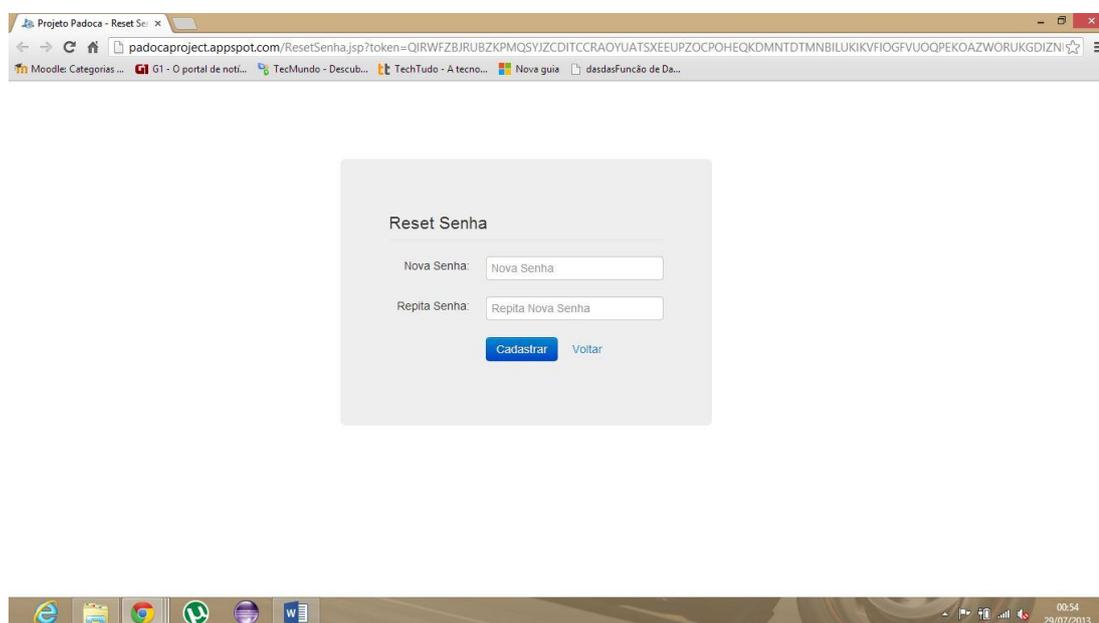
Caso o usuário tenha esquecido a sua senha ou esteja com seu login bloqueado pelo sistema, ele poderá utilizar a opção “Esqueci minha senha” presente na tela de login. Essa opção levará o usuário para a tela de recuperação de senha, apresentada na Figura 6.



**Figura 6 – Tela de Recuperação de Senha**

O usuário informará o e-mail utilizado para cadastro no sistema, que validará a existência desse e-mail. Caso seja válido, será enviado um comunicado para o e-mail

do usuário com um link para a página de reset de senha, apresentada na Figura 7, onde o usuário cadastrará a sua nova senha de acesso.

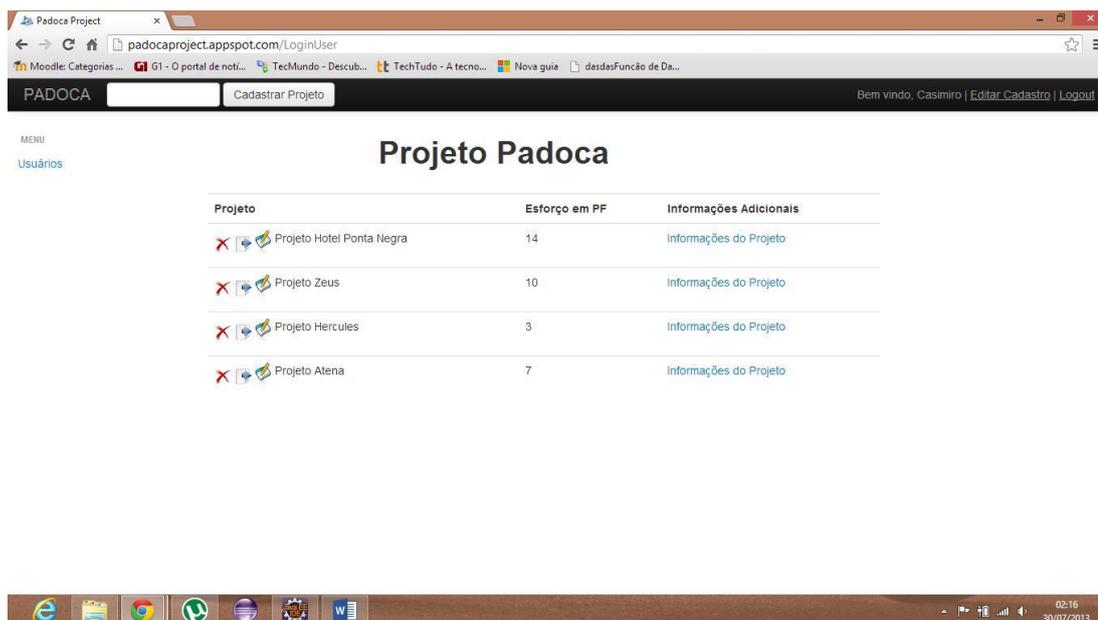


**Figura 7 – Tela de Reset de Senha**

Após cadastrar a nova senha, o usuário será encaminhado novamente para a tela de login do sistema (Figura 4), onde poderá fazer o acesso com a nova senha. Todos as telas desse processo possuem as validações necessárias para garantir a integridade e segurança dos dados armazenados no sistema, como validações sobre o preenchimento dos campos, garantia de que não existam logins duplicados na base e de que o login informado para reset de senha esteja cadastrado na base. Caso alguma dessas validações falhe, serão apresentadas mensagens de erro para o usuário.

## **4.2 - Funcionalidades do Sistema**

Ao acessar o sistema o usuário será encaminhado para a sua tela principal, onde estão listados todos os sistemas cadastrados por ele, apresentando seu nome, número de pontos de função e um link para acessar as informações adicionais do projeto. A Figura 8 apresenta a tela inicial do sistema.

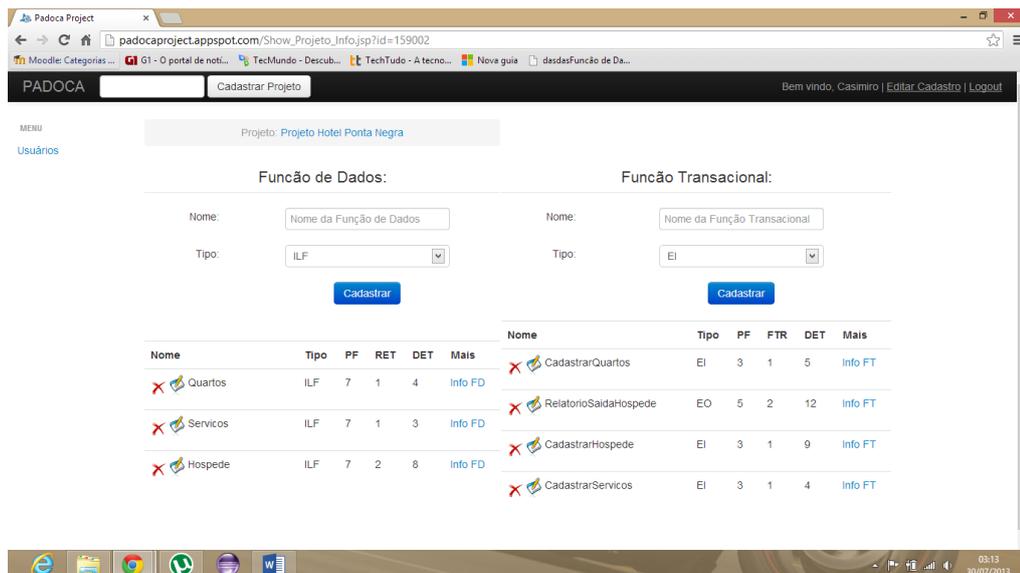


**Figura 8 – Tela Inicial do Sistema**

Cada projeto listado na tela principal apresentará ícones do lado esquerdo que acionam determinadas funcionalidades para esse projeto, como: exclusão do projeto, exportação das informações do projeto para um formato XML pré-definido e alteração dos dados cadastrais do projeto.

Além disso, na parte superior da tela é apresentada uma barra que possui algumas funcionalidades do sistema. Nessa barra estão disponíveis as funcionalidades de cadastro de um novo projeto, edição das informações cadastrais do usuário logado e logout do sistema. As funcionalidades presentes nessa barra poderão ser acessadas em qualquer janela do sistema, tendo como única condição que o usuário esteja logado.

Ao selecionar a opção “Informações do Projeto” de algum dos projetos, o sistema levará o usuário para uma página com um detalhamento das funções previstas para o projeto. Essa tela é apresentada na Figura 9. Nela são exibidos dois formulários de cadastro. O primeiro se refere às funções de dados, contendo campos para seleção do nome e do tipo da função (que pode ser um ILF ou EIF). O segundo formulário é para cadastro das funções transacionais, contendo os campos nome e tipo da função transacional (que pode ser um EI, EO ou EQ).



**Figura 9 – Tela de Informações do Projeto**

Abaixo de cada um dos formulários está a lista com as funções de cada tipo cadastradas para o projeto. Para as funções de dados essa lista possui os seguintes campos: nome, tipo, esforço em pontos de função, quantidade de RETs e DETs, além de um link para o detalhamento das informações daquela função de dados. Para as funções transacionais a lista possui os seguintes campos: nome, tipo, esforço em pontos de função, quantidade de FTRs e DETs e um link para o detalhamento das informações daquela função de dados. Nas duas listas estão disponíveis dois botões, sendo o primeiro para exclusão da função de dados ou transacional escolhida e o segundo para edição no nome dessa função.

Ao acessar a opção “Info FD” na lista de funções de dados cadastradas para o projeto (Figura 8), o sistema exibirá uma tela com o detalhamento dessa função de dados. Essa tela é apresentada na Figura 10. Ela possui um formulário para cadastro de um atributo na função de dados. Esses atributos possuem um nome e o tipo, podendo ser um RET ou DET. Ao selecionar o tipo DET, o formulário apresentará uma caixa de seleção para que seja informado a que RET aquele DET pertence. Abaixo desse formulário estão listados todos os RETs que compõem uma função de dados, o número de DETs que cada um possui e uma lista com o nome desses DETs. Nessa lista também é apresentado um botão para a exclusão de algum DET pertencente ao RET ou do próprio RET como um todo.

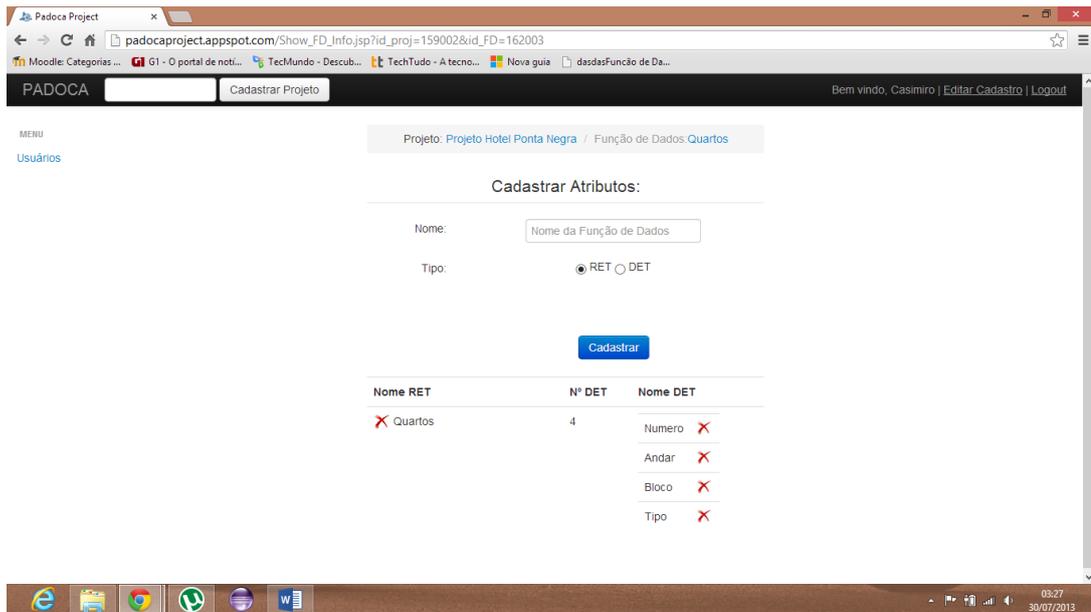


Figura 10 – Tela de Informações da Função de Dados

Ao acessar a opção “Info FT” na lista de funções transacionais cadastradas para o projeto (Figura 9), o sistema exibirá uma tela com o detalhamento dessa função transacional. Essa tela é apresentada na Figura 11. Ela possui um formulário para cadastro de um atributo na função transacional. Esse formulário possui um campo “Nome FD” e o campo “Nome DET”. No campo “Nome FD” estão listadas todas as funções de dados cadastradas no projeto e o campo “Nome DET” listará todos os DETs cadastrados para aquela função de dados.

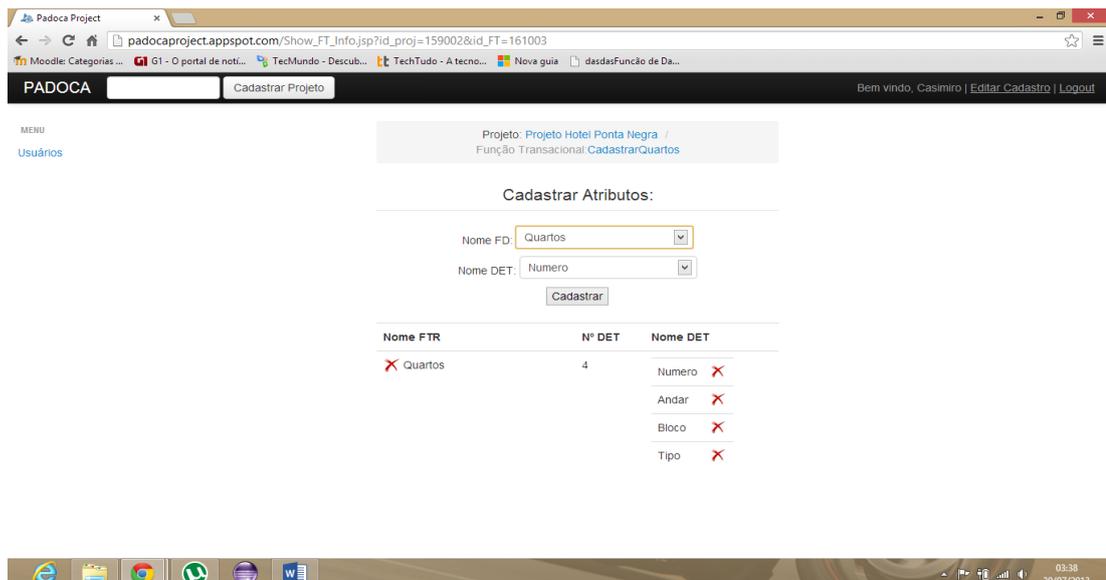


Figura 11 – Tela de Informações da Função Transacional

Para cadastrar um FTR é necessário que sejam selecionados a função de dados e o DET que o irão compor, visto que um FTR precisa ser cadastrado com pelo menos um DET. Após isso, o usuário poderá atribuir mais DETs para aquele FTR cadastrado, sendo a única condição que o DET esteja cadastrado na função de dados que aquele FTR representa. Abaixo do formulário de cadastro estão listados todos os FTRs que compõem uma função transacional, o número de DETs que cada FTR possui e uma lista com o nome desses DETs. Nessa lista também é apresentado um botão para a exclusão de algum DET pertencente a um FTR ou do próprio FTR.

Após cadastradas todas as funções de dados e transacionais de um projeto, incluindo seus atributos, o usuário poderá verificar o número de pontos de função calculado para o projeto voltando à tela inicial, onde na linha que representa o projeto será exibido seu número de pontos de função, conforme demonstrado na Figura 12.

Projeto	Esforço em PF	Informações Adicionais
 Projeto Hotel Ponta Negra	35	<a href="#">Informações do Projeto</a>

**Figura 12 – Demonstração do número de pontos de função de um projeto**

O cálculo de ponto de função é calculado de forma independente para cada função cadastrada em um projeto. Para cada função o sistema analisará o seu tipo, ILF ou EIF caso seja uma função de dados e EI, EO e EQ caso seja uma função transacional, com essa informação fará o cálculo da quantidade de RETs, para as funções de dados ou FTRs para as transacionais, e DETs presentes nessa função, descobrindo assim a complexidade da mesma através das regras expostas anteriormente. Após essa análise, de posse do tipo de função e da sua complexidade, o sistema irá atribuir um número de pontos de função para aquela função de acordo com as regras definidas para cada tipo de função. Após o cálculo da contribuição em pontos de função para todas as funções cadastradas, o sistema fará a soma de cada um desses valores, apresentando o resultado encontrado para o usuário final, sendo esse resultado o tamanho do projeto representado em pontos de função.

Os projetos apresentados nesta lista ainda podem ser exportados para um documento em formato XML, conforme demonstrado na Figura 13.

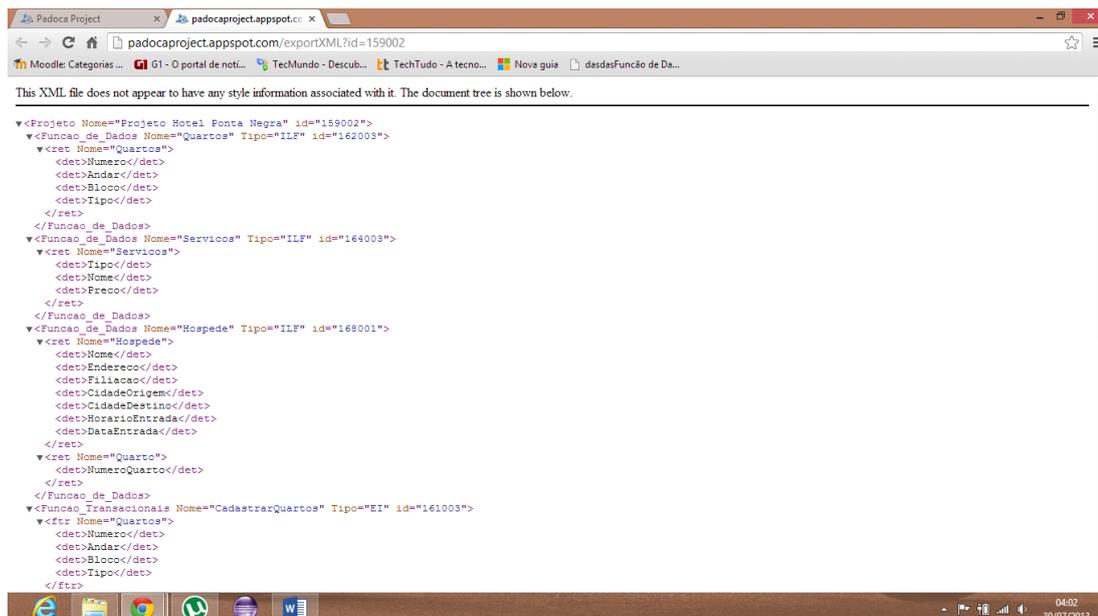


Figura 13 – Tela de Exportação do projeto em formato XML

E por fim, é apresentada a opção de edição do nome do projeto (também presente para a edição do nome de funções de dados ou transacionais). Um exemplo da tela de edição é apresentado na Figura 14.

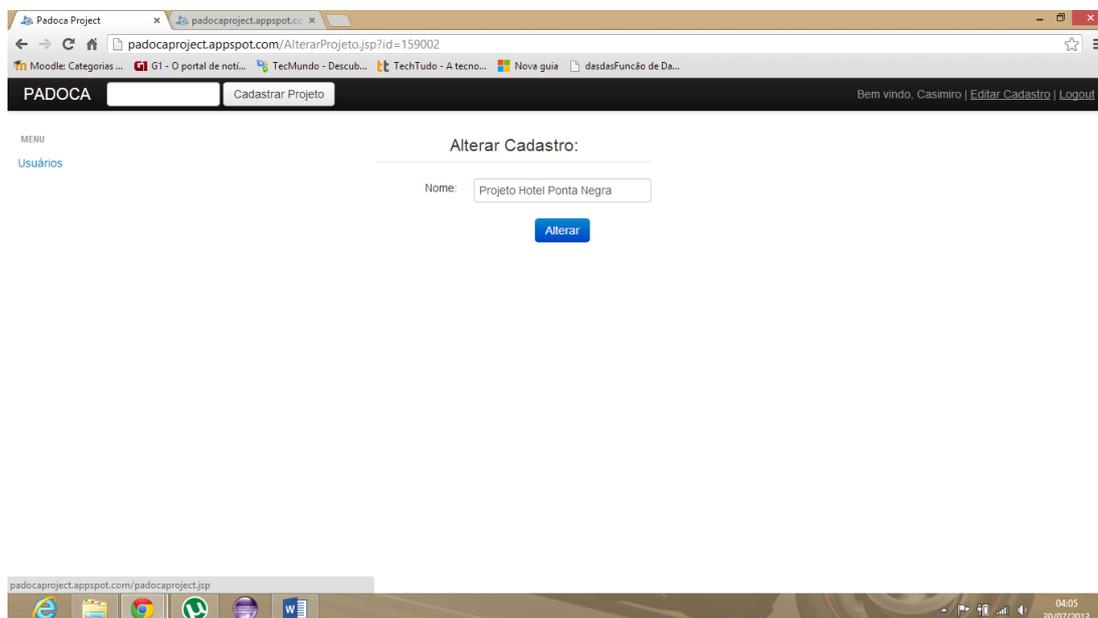


Figura 14 – Tela de Edição de Informações

### 4.3 Considerações Finais

Nesse capítulo foram apresentadas as telas presentes no sistema de contagem de pontos de função e um resumo do fluxo de utilização da ferramenta. No próximo

capítulo apresentaremos a conclusão do projeto, apresentando um resumo do observado durante seu desenvolvimento, as possíveis evoluções em trabalhos futuros e as limitações do projeto atual.

## **5 Conclusões**

Esse capítulo conclui o projeto de desenvolvimento de uma ferramenta para cálculo de esforço em pontos de função. Nele será apresentado um resumo do que foi observado durante o projeto, as possíveis evoluções futuras e as limitações da versão atual do projeto.

### **5.1 - Observações Finais**

Uma métrica alcança seu sucesso e objetivo quando ela auxilia a tomada de ação das áreas gestoras. Esse sucesso depende de diversos fatores, dentre eles a conscientização e existência de recursos humanos e tecnológicos que possibilitem essa medição. As métricas trazem diversos benefícios a uma empresa, visto que possibilitam o dimensionamento das atividades, carga de trabalho, qualidade, entre outros, para assim garantir a assertividade dos projetos e atividades mediadas.

Ponto de Função é uma técnica muito interessante para utilização na medição do esforço para o desenvolvimento de um software, visto que tira o foco da tecnologia que será empregada, do ambiente e demais fatores, para focar sua análise nas funcionalidades do ponto de vista dos usuários solicitantes.

Assim a Análise de Pontos de Função pode ser usada para a definição de metas e indicadores de esforço, que serão alimentados com o histórico de projetos executados pela unidade e, dessa forma, ajudará a identificar possíveis problemas no processo, baseando-se, por exemplo, na taxa de produtividade definida para determinada unidade. Sendo assim, é um grande auxílio na tomada de decisão e sendo usada da maneira correta, poderá beneficiar a cadeia produtiva da área de TI.

De forma análoga, as tecnologias, Google App Engine e NoSQL, utilizadas no desenvolvimento do projeto se mostraram de grande ajuda e interferiram positivamente na qualidade do resultado apresentado.

## **5.2 - Evoluções Futuras**

O sistema implementado pelo projeto é apenas a fase inicial de uma ferramenta que ainda pode evoluir muito, acrescentando funcionalidades muito interessantes para o acompanhamento e gerência de projetos de software.

Entre essas funcionalidades estão as possibilidades de integração com ferramentas de modelagem através do XML gerado (ver seção 4.2), podendo ser implementadas novas funcionalidades de importação de projetos a partir de um modelo e exportação desse projeto para as ferramentas de modelagem.

Outra possível evolução é a criação de um cronograma macro de um projeto, baseado nas atividades-chave do processo de desenvolvimento de uma organização e no cálculo de Pontos de Função feito pelo sistema. Também é possível criar novas funcionalidades de cálculo, em que o número de Pontos de Função do projeto seja convertido em estimativas de interesse da organização, como estimativa de custo, de tempo e qualidade.

Por fim, outra funcionalidade possível seria a criação de um histórico para acompanhamento da evolução que o cálculo de Pontos de Função pode vir a sofrer durante o andamento do projeto.

Essas e outras funcionalidades podem ser agregadas ao sistema, tornando-o mais completo e maduro para possível utilização em mercado.

## **5.3 - Limitações do Projeto**

Sendo essa a sua primeira versão, o sistema apresenta algumas outras limitações, como: (i) o sistema não possui um log de alterações feitas em cada projeto; (ii) o sistema não possui a funcionalidade de importação de um projeto em formato XML; e (iii) o sistema não possui perfis de acesso, visto que as funcionalidades previstas no escopo do projeto não requeriam a implementação de um acesso diferenciado por perfis de usuário.

# Referências Bibliográficas

VARGAS, R.V.; Gerenciamento de Projeto, estabelecendo diferenciais competitivos. ISBN 85-7452-208-2. Ed. Atual, Rio de Janeiro, 2005. Disponível (também) em: <[http://books.google.com.br/books?id=Wvdk7IaOC7wC&printsec=frontcover&hl=pt-BR&source=gbs\\_ge\\_summary\\_r&cad=0#v=onepage&q&f=false](http://books.google.com.br/books?id=Wvdk7IaOC7wC&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false)>, acessado em 01/07/2013.

DIAS, R.; Análise por Pontos de Função: Uma Técnica para Dimensionamento de Sistemas de Informação, 2003. Disponível (também) em: <<https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CC4QFjAA&url=http%3A%2F%2Frevistas.facecla.com.br%2Findex.php%2Freinfo%2Farticle%2Fdownload%2F134%2F28&ei=pyj7UamvNZPa9ASP1IBY&usg=AFQjCNGoNVyr9rTPgShT3m9rVgzNrp4oyw>>, acessado em 01/07/2013.

PMI, Project Management Body of Knowledge – Fifth Edition, ISBN: 978-1-933890-70-8. Project Management Institute, Pennsylvania, USA, 2012

IFPUG; Function Point Counting Practices Manual - Release 4.3, ISBN 978-0-9753783-3-5. IFPUG, NJ, USA, 2010.

SOUZA, T.; ROCHA, A.; NoSQL Princípios e Características, 2010. Disponível (também) em <<http://www.slideshare.net/andrerochajp/artigo-nosql>>, acessado em 27/07/2013.

MELL, P.; GRANCE, T.; The NIST Definition of Cloud Computing, Disponível em <<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>>, 2011, acessado em 27/07/2013.

ORACLE; The History of Java Technology, 2011. Disponível (também) em: <<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index198355.html>>, acessado em 27/07/2013.

IBM; A Brief History of Eclipse, 2005. Disponível em: <<http://www.ibm.com/developerworks/rational/library/nov05/cernosek/index.html>>, acessado em 27/07/2013.

GOOGLE; Google Plugin for Eclipse, 2013. Disponível em: <<https://developers.google.com/eclipse/?hl=pt-BR>>, acessado em 27/07/2013.

ZYP, K.; NoSQL Architecture, 2010. Disponível (também) em: <<http://www.sitepen.com/blog/2010/05/11/nosql-architeceu-ature/>>, acessado em 29/07/2013

STONEBRAKER, MIKE, et al; C-Store: A Column-oriented DBMS, 2005.  
Disponível (também) em: <<http://db.lcs.mit.edu/projects/cstore/vldb.pdf>>, acessado  
em 29/07/2013