

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DE SÃO PAULO**

Campus São João da Boa Vista

Trabalho Final de Curso

4º ano – Curso Técnico em Informática

Prof. Breno Lisi Romano

TÍTULO

Aluno: Letícia dos Anjos da Silva Araújo

Prontuário: 1420216

São João da Boa Vista – SP

2017

Resumo

Em grandes projetos há muita dificuldade em estabelecer um valor. A estimativa de esforços por Pontos de Caso de Uso é o método mais recomendado pela engenharia de software para se calcular o valor de um sistema. O objetivo deste trabalho é explorar essa técnica utilizando uma ferramenta própria para a realização desse cálculo, sem ser necessária experiência, apenas conhecimento dos passos de realização do devido projeto, obtendo resultados coerentes com os fatores de esforços.

Palavras-chave: Estimativa. Custo. PCU. Esforço. Engenharia de Software.

Sumário

1	Introdução	4
2	Desenvolvimento	6
	2.1 Referencial teórico	6
	2.2 Metodologia.....	8
	2.2.1 Diagrama de Caso de Uso	8
	2.2.2 Peso dos Atores	9
	2.2.3 Peso dos Casos de Uso	11
	2.2.4 Peso Total.....	12
	2.2.5 Fatores de Ajuste: Fatores Técnicos.....	13
	2.2.6 Fatores de Ajuste: Fatores Ambientais.....	17
	2.2.7 Pontos de Caso de Uso	20
	2.2.8 Estimativa de Tempo de Trabalho	20
	2.2.9 Custo de Desenvolvimento.....	21
	2.2.10 Estimativas de Tempo	22
	2.2.11 Estimativa do Projeto Reclame São João	24
3	Referências Bibliográficas	27

1 Introdução

O tema desse trabalho é a estimativa de esforços de software e a forma como foi aplicada no módulo de reclamações do projeto Reclame São João de 2017. Para iniciar o desenvolvimento do projeto, são feitas documentações dentro de cada módulo. Esse método permite que seja estimado o tamanho do projeto quando ele está em sua fase de levantamento dos Casos de Uso, ou seja, utiliza-se a própria documentação. Todos os registros feitos nesse documento são utilizados para obter a estimativa de esforços do software.

O projeto Reclame São João foi aberto no dia 20 de fevereiro de 2017, com um Termo de Abertura no qual havia um campo a ser preenchido, chamado Orçamento. Esse orçamento foi dado através de uma estimativa não muito precisa, baseada apenas no piso salarial de um profissional de técnico em informática júnior trabalhando em empresas de médio porte, e 20% de aumento que inclui custo de transportes, energia utilizada e a inflação. Por que é necessário um programa que meça os esforços num software para valorizá-lo, invés de estabelecer o preço que se deseja fundamentando-se em fatores como o piso salarial de um profissional, por exemplo?

Esse tema será estudado porque é muito comum que não se saiba como agregar valor a um grande projeto. Há muitos softwares que são considerados caros, por exemplo, o AutoCAD da desenvolvedora Autodesk que o é de 4.415,49 reais por ano. O preço pode ser considerado caro pelos clientes, mas é preciso compreender que todo programa tem um custo de produção, e esse trabalho tem como objetivo demonstrar como medir esse custo.

Será utilizada a pesquisa descritiva, experimental e exploratória para apresentar a ferramenta utilizada pelo Reclame São João e explicar todas as etapas que o módulo de Reclamações do projeto passou para chegar ao seu orçamento final através dela; a EEDS-PCU (Estimativa de Esforço de Desenvolvimento de Software com Ponto de Casos de Uso), o software que foi desenvolvido por Caio Monteiro Barbosa da Silva em sua pós-graduação em Engenharia Eletrônica e Computação na Área de Informática no ITA (Instituto Tecnológico de Aeronáutica) no ano de 2006, atendendo as orientações do Professor Doutor Adilson Marques da Cunha da Divisão da Ciência da Computação do ITA.

Este trabalho está organizado em quatro capítulos, contendo a introdução, na qual, é apresentado o tema, a motivação, os objetivos, e a metodologia da pesquisa; o desenvolvimento, que contém o referencial teórico de todo o estudo da estimativa de esforços pela ferramenta EEDS e a demonstração da aplicação no módulo de Reclamações; a conclusão, que irá expor a importância

da estimativa de esforços baseando-se nos resultados obtidos através da pesquisa; e por fim, referências bibliográficas, que irá conter toda a bibliografia de estudo do trabalho.

2 Desenvolvimento

2.1 Referencial teórico

O tamanho do produto de software tem sido apontado como um ponto de partida para a realização de estimativas [1]. Nos anos 90 estavam sendo feitas pesquisas que mostravam como medir essa grandeza, principalmente a PCU (Análise de Pontos de Caso de Uso), e a APF (Análise de Pontos de Função).[2] Após esses estudos, foi concluído que a APF não era o método adequado para se medir sistemas orientados a objetos, uma vez que só pode ser usada para grandes projetos, e além de existirem muitas variantes do método, as faixas de complexidade são de difícil adaptação a realidade em muitos casos, da mesma forma que sua contagem não pode ser automatizada com facilidade [1], então pode se dizer que é uma técnica inviável para o tipo de projeto que ela atende.

Considerando essas conclusões, em 1993, foi proposta a metodologia da PCU, baseada na APF, em um trabalho acadêmico por Gustav Karner. Essa tem o objetivo de estimar recursos para projetos de software orientados a objeto. O processo de medição de Pontos de Caso de Uso consiste em contar os atores e casos de uso e identificar sua complexidade, calcular os PCUs não ajustados e ajustados e determinar o fator de complexidade técnica e ambiental; para tornar possível a estimativa de esforço total para o projeto. [3]

As documentações de Casos de Uso, que são utilizadas para a estimativa de esforços, partem de um diagrama, que contém como elementos os Atores, os Casos de Uso, Interações e Associações e as Fronteiras de Sistema [4]. Esse traz melhor visualização do programa, que permite que desenvolvedores de software discutam suas funcionalidades e sejam guiados para o processo do desenvolvimento, mostrando o que se deve fazer para que decidam como.

Os Atores são qualquer elemento externo que interaja de alguma forma com o sistema. Isto é, não fazem parte desse mas enviam e recebem informações dele. Por exemplo, usuários; desenvolvedores que mantêm e iniciam e encerram o sistema; e até outros sistemas que são utilizados ou utilizam-no. No diagrama são representados por um boneco rotulado com seu nome.

Os Casos de Uso são interações típicas entre usuário e programa a partir de um ponto de vista segmentado, ou seja, um conjunto de sequência de ações que um sistema desempenha para produzir um resultado observável de valor a um ator específico [5]. No diagrama são representados por uma elipse rotulada com o nome relacionado. São sempre iniciados pelo Ator, sendo direta ou indiretamente. Quando o Ator está associado com o Caso de Uso, isso quer dizer que está sendo definida uma das funcionalidades do sistema do ponto de vista do usuário. Há casos em que dois atores estão relacionados entre si, o fenômeno é chamado de generalização, quando os Casos de Uso de um Ator são também de outro, sendo que aquele tem Casos de Uso próprios. No diagrama

essas associações são representadas por setas, seja de Ator para Caso de uso, ou de Ator para Ator. A interação chamada generalização incide entre Casos de Uso também, permitindo que um Caso de Uso herde características de outro mais genérico, quando têm comportamentos semelhantes [5].

Quando os relacionamentos são entre os Casos de Uso, há dois tipos de interações, sendo elas *Include* (inclusão) e *Extend* (extensão). Um relacionamento nomeado *Include* é utilizado quando Casos de Uso têm comportamento em comum ou indicam obrigatoriedade, isto é, a execução de um deles implica na execução do outro. No diagrama a Inclusão é figurada com uma seta de linha tracejada e estereotipada como <<include>> do Caso de Uso base apontando para o que ele implica. Enquanto o relacionamento nomeado *Extend* é ocorrente para indicar comportamento de exceção e casos especiais, indicando um comportamento opcional, ou seja, cursos alternativos que na minoria dos casos ocorrem. No diagrama a Extensão é figurada com uma seta de linha tracejada e estereotipada como <<extend>> do Caso de Uso alternativo apontando para o Caso de Uso principal [5].

As fronteiras de sistema representam ambientes diferentes dentro de um mesmo programa, ou seja, subsistemas. São ilustradas com retângulos que, no diagrama, envolvem os Casos de Uso a que eles dizem respeito. Numa questão de organização e melhor visualização destacam o que está contido naquele subsistema do que não está.

Visto isso, é possível calcular a complexidade do software para a estimativa de esforços com os dados dos Pontos de Casos de uso. Victorio A. de Carvalho, Lucas O. Arantes, Ricardo de Almeida Falbo apresentam em seu trabalho EstimaODE: Apoio a Estimativas de Tamanho e Esforço no Ambiente de Desenvolvimento de Software ODE, de 2006, [6] outras formas de estimativas, pois para garantir a realização de boas estimativas, é aconselhável que o gerente de projeto possua experiência, utilize diferentes abordagens para realizar estimativas, incluindo dados históricos, e pondere complexidade e riscos [7]. Contudo, o objetivo a ser cumprido com essa pesquisa é apresentar esse método, que foi utilizado na realização do projeto Reclame São João.

2.2 Metodologia

2.2.1 Diagrama de Caso de Uso

Após o levantamento de requisitos funcionais e não funcionais do módulo Reclamações do projeto Reclame São João, foi feito o diagrama de Casos de Uso, utilizando a ferramenta StarUML. Como atores foram identificados o Consumidor, o Google Maps e o Módulo 02 (Reclamações).

O Consumidor é um usuário de extrema importância em Reclamações, pois ele é responsável pela maioria das ações, uma vez que esse realiza reclamações sobre os estabelecimentos registrados na página. Para realizar uma reclamação é necessário que se pesquise um estabelecimento existente, o Caso de Uso referente a essa ação é o “pesquisar e exibir estabelecimento” [8].

Ao encontrar a empresa desejada o Consumidor pode reclamar sobre ele, exibir sua reputação ou avalia-lo. Essas ações são referentes, respectivamente, aos Casos de Uso “criar nova reclamação”, “exibir reputação” e “avaliar estabelecimento”. Sendo que essas são extensões de “pesquisar e exibir estabelecimento” [8].

Se o usuário já possui reclamações feitas, ele pode gerenciá-las, ou seja, excluí-las ou editá-las. Essas ações são referentes, respectivamente, aos Casos de Uso “gerenciar reclamações”, “excluir reclamação” e “editar reclamação”. Sendo que essas são extensões de “gerenciar reclamações” [8].

Ele também é notificado quando uma reclamação feita é respondida pela empresa reclamada, se trata de um Caso de Uso chamado “receber notificação resposta da reclamação” [8], sem extensões ou inclusões.

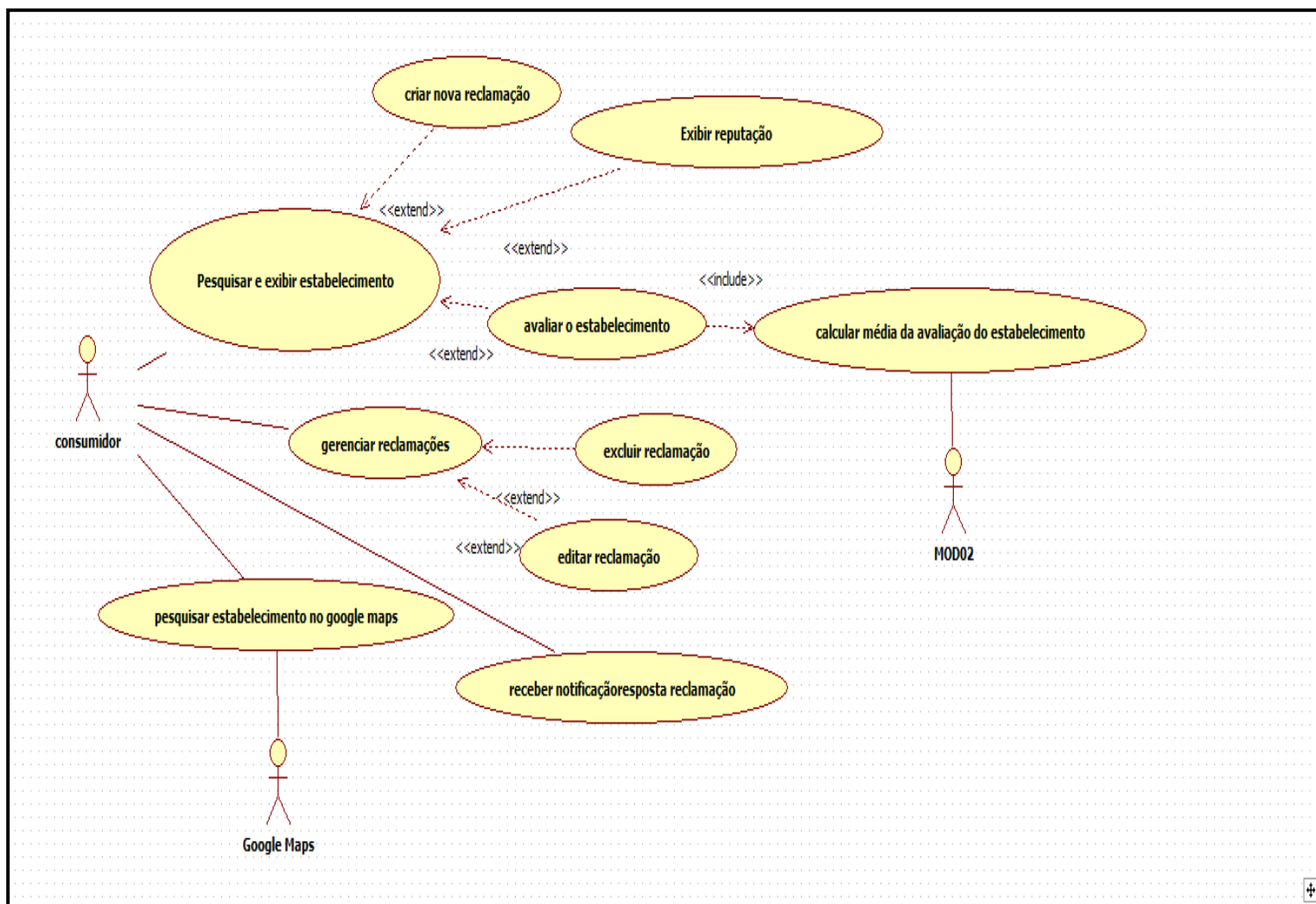
A última ação feita diretamente pelo Consumidor, é pesquisar o estabelecimento desejado na ferramenta do Google Maps no site, esse é o Caso de Uso nomeado “pesquisar estabelecimento no Google Maps” [8], que também não possui extensões ou inclusões.

O ator MOD02 é responsável pelo Caso de Uso “calcular média da avaliação do estabelecimento” que é uma inclusão do Caso de Uso “avaliar estabelecimento” [8] porque sempre que um Consumidor faz uma nova avaliação, o módulo Reclamações deve garantir que a média de avaliações do estabelecimento seja atualizada.

O ator Google Maps é um sistema utilizado pelo Reclame São João que exibe todas as empresas registradas no mapa da cidade em seu endereço correto, ele é responsável também pelo Caso de Uso “pesquisar estabelecimento no Google Maps” [8] pois é o sistema a ser consultado para que essa ação seja realizada por outro usuário, no caso, o Consumidor.

Então diagrama de Caso de Uso final a ser utilizado nessa fase de análise como subsídio para o cálculo dimensional obtido pelo módulo 02 foi este:

Figura 01 – Diagrama de Casos de Uso MOD02



Fonte: Módulo 02 – Reclamações, Modelo de Casos de Uso, V.2.3

2.2.2 Peso dos Atores

O primeiro passo para a estimativa foi o cálculo do peso dos Atores presentes no diagrama. Os Atores relacionados a cada Caso de Uso foram classificados para obter a soma dos pontos. O nome dado para isso é UAW (Anadjusted Actor Weight) [3], é calculado através da soma dos produtos da quantidade de atores de cada grau de complexidade pelo seu peso. A tabela abaixo explica os tipos de atores e seus pesos, e a aplicação no diagrama do Módulo 02; a tabela seguinte mostra como foi realizado o cálculo e a imagem subsequente contém a interface do EEDS, que realiza o cálculo automaticamente, com o primeiro passo feito.

Tabela 01: Atores MOD02

Tipo de Ator	Descrição	Peso	Aplicação
Simple	Outro sistema que é acessado por um API de programação	1	Google Maps
Médio	Outro sistema que é acessado interagindo pela rede	2	MOD02
Complexo	Usuário interagindo diretamente com o sistema pela interface gráfica	3	Consumidor

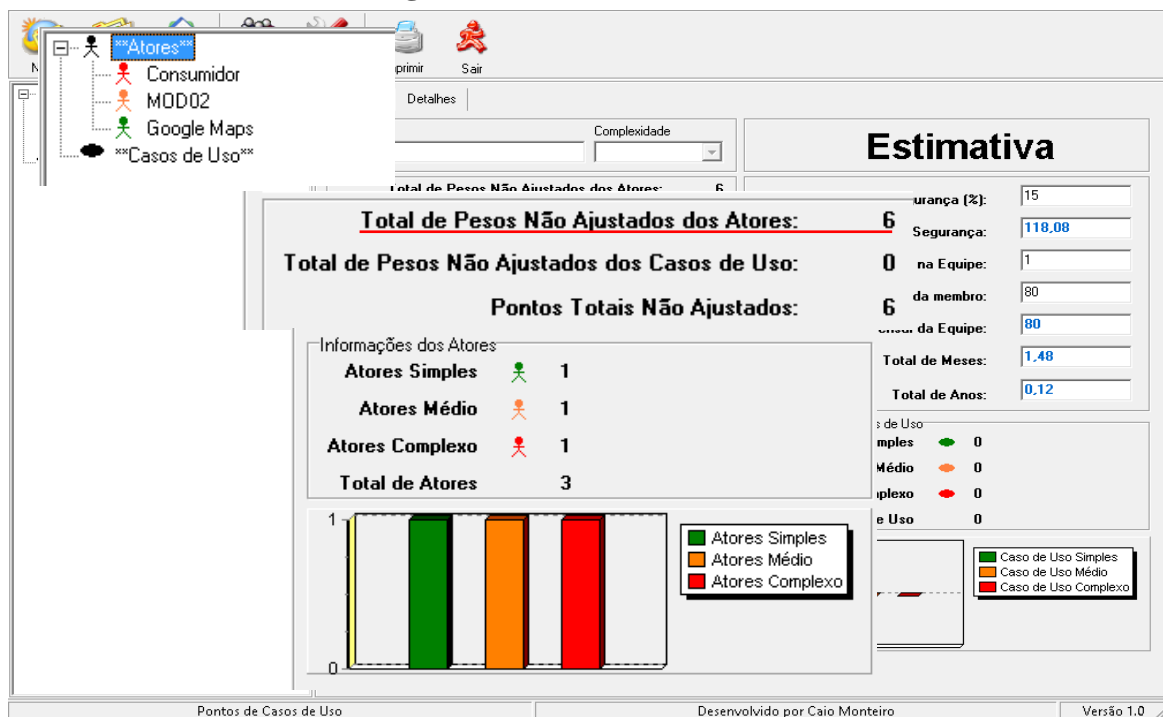
Fonte: Elaboração própria

Tabela 02: Cálculo do Peso dos Atores MOD02

Tipo de Ator	Peso	Quantidade de Atores	Resultado
Simple	1	1	1
Médio	2	1	2
Complexo	3	1	3
UAW total			6

Fonte: Elaboração própria

Figura 02 – Peso dos Atores MOD02



Fonte: Elaboração própria

2.2.3 Peso dos Casos de Uso

O segundo passo foi o cálculo do peso dos Casos de Uso presentes no diagrama. Foram classificados para obter a soma dos pontos. O nome dado para isso é UUCW (Unadjusted Use Case Weight) [3], é calculado através da soma dos produtos da quantidade de Casos de Uso de cada grau de complexidade pelo seu peso. A tabela abaixo explica os tipos de Casos de Uso e seus pesos, e a aplicação no diagrama do Módulo 02; a tabela seguinte mostra como foi realizado o cálculo e a imagem subsequente contém a interface do EEDS, que realiza o cálculo automaticamente, com o segundo passo feito.

Tabela 03: Casos de Uso MOD02

Tipo de Caso de Uso	Descrição	Peso	Aplicação
Simples	Possui no máximo 3 transações	5	Pesquisar estabelecimento no Google Maps; Receber notificação resposta reclamação; Gerenciar Reclamações; Excluir reclamação; Editar reclamação; Cálculo média avaliação estabelecimento; Exibir reputação; Criar nova reclamação; Avaliar estabelecimento
Médio	Possui de 4 a 7 transações	10	Pesquisar e exibir estabelecimento
Complexo	Possui no mínimo 7 transações	15	-

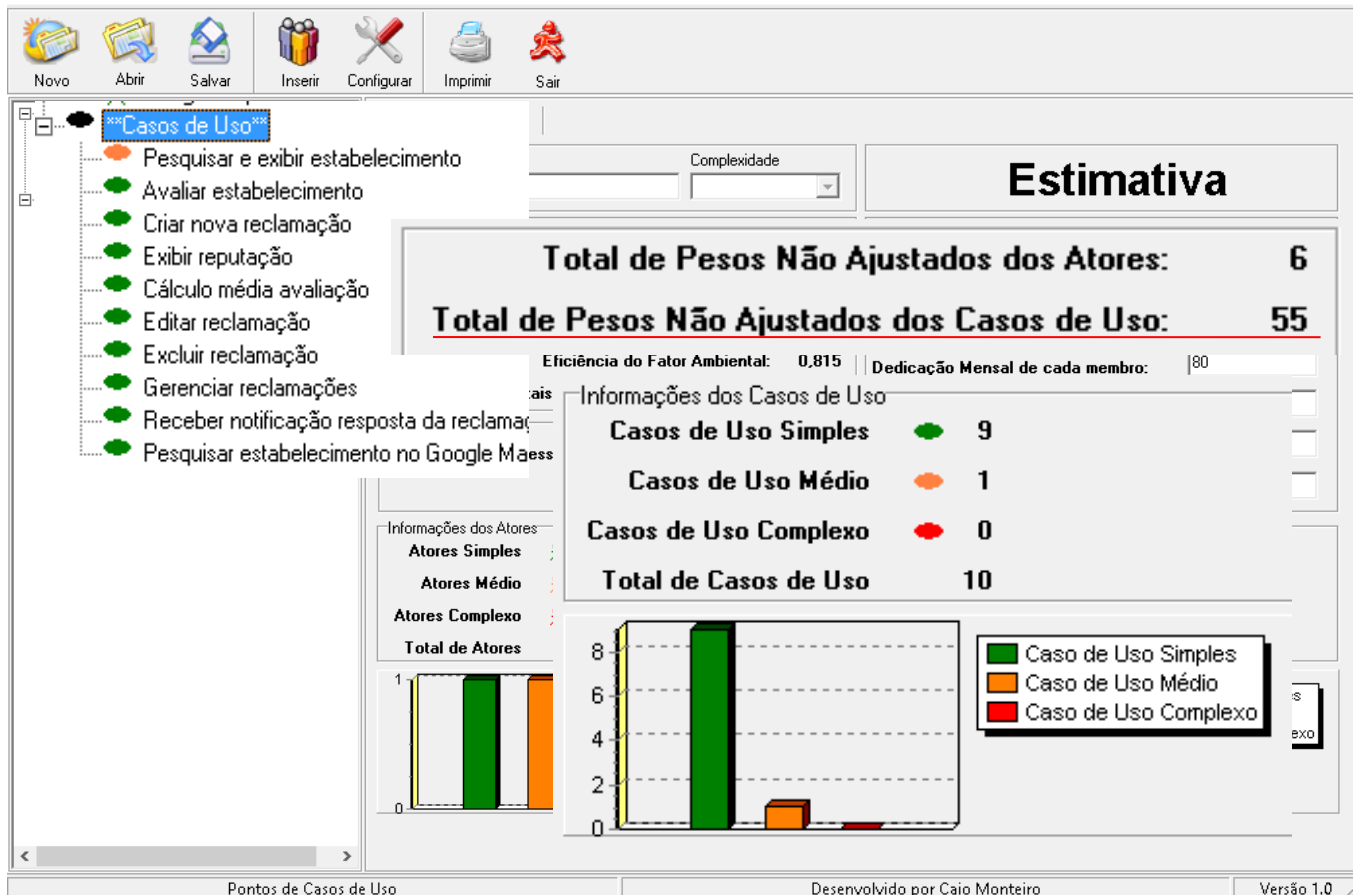
Fonte: Elaboração própria

Tabela 04: Cálculo do Peso dos Casos de Uso MOD02

Tipo de Caso de Uso	Peso	Quantidade de Casos de Uso	Resultado
Simples	5	9	45
Médio	10	1	10
Complexo	15	0	0
UUCW total			55

Fonte: Elaboração própria

Figura 03 – Peso dos Casos de Uso MOD02



Fonte: Elaboração própria

2.2.4 Peso Total

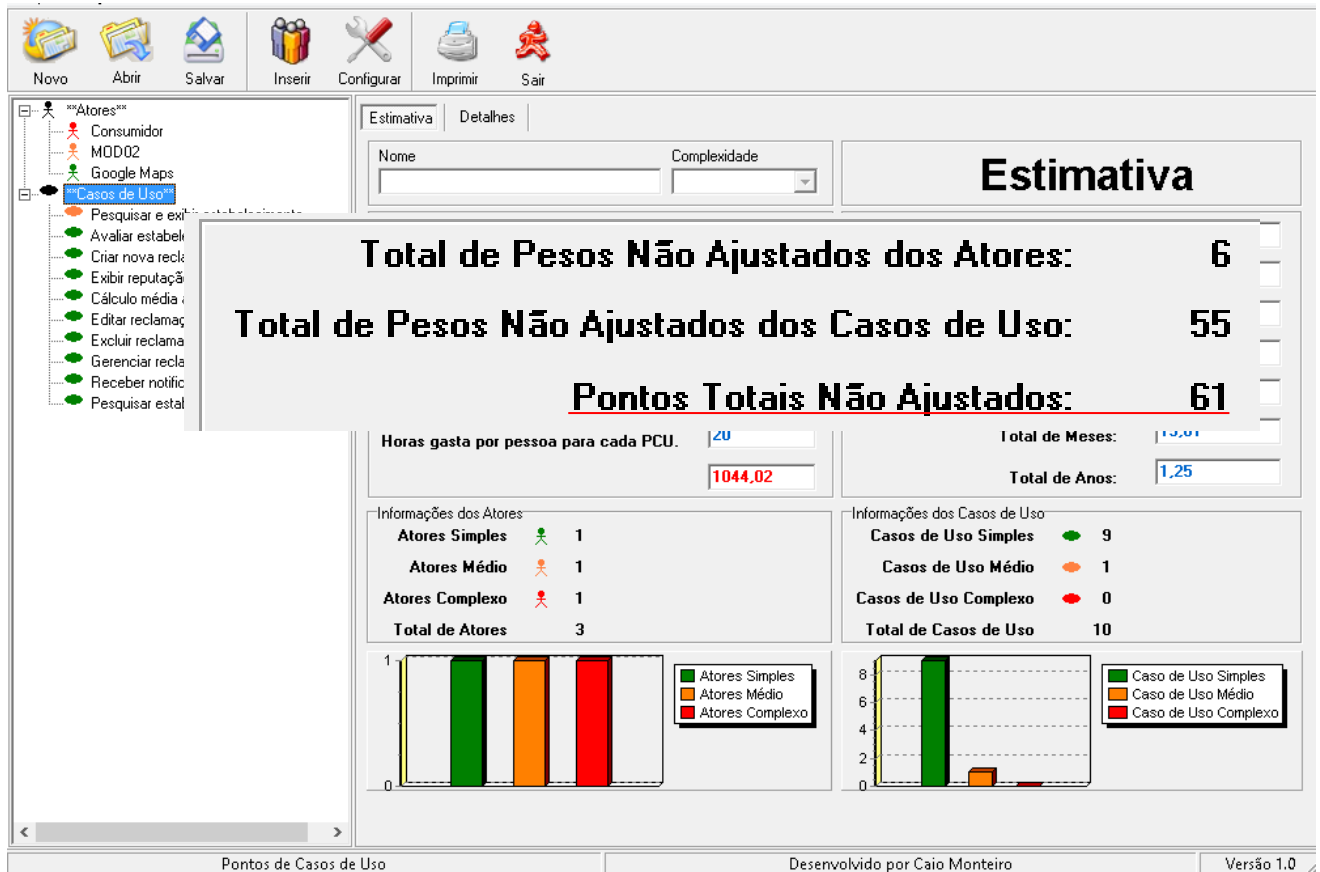
O terceiro passo foi o cálculo do peso total no diagrama. O nome dado para isso é UUCP (Unadjusted Use Case Points) [3], é calculado através da soma do resultado o UAW com o UUCW. A tabela abaixo explica como foi realizado o cálculo e a imagem seguinte contém a interface do EEDS, que realiza o cálculo automaticamente, com o terceiro passo feito.

Tabela 05: Cálculo do Peso Total MOD02

UAW	UUCW	UUCP
6	55	61

Fonte: Elaboração própria

Figura 04 – Peso Total MOD02



Fonte: Elaboração própria

2.2.5 Fatores de Ajuste: Fatores Técnicos

Em seguida, foram calculados os Fatores de Ajuste da estimativa. O Fator de Ajuste pode enriquecer ou empobrecer os PCU, em função dos valores de 0 a 5, a respeito da influência do requisito no sistema atribuídos a ele. Sendo que cada requisito tem um peso para o cálculo final.

Primeiramente foi feito o cálculo de fatores técnicos, também chamados de Tfactor [3], que são aqueles que representam requisitos funcionais do sistema. Esses são:

- Sistema distribuído: se há CPU apropriada para as funções de processamento;
- Desempenho da aplicação: se ações especiais de performance foram tomadas;
- Eficiência de usuário final: se funções online fornecidas enfatizam um projeto da aplicação voltado para a eficiência do usuário final, ex: menus, documentação/Help on-line, seleção de dados da tela via movimentação do cursor, Movimento automático do cursor, movimento de tela vertical e horizontal, pop-up Windows;
- Complexidade de processamento interno: dividido nas categorias processamento especial de auditoria e/ou processamento especial de segurança; processamento

lógico extensivo; processamento matemático extensivo; grande quantidade de processamento de exceções, resultante de transações incompletas que necessitam de reproprocessamento, por exemplo: transações incompletas de ATMs causadas por interrupções de comunicação, valores de dados ausentes ou validações de erros; processamento complexo para manipular múltiplas possibilidades de entrada/saída, por exemplo: múltiplos meios e independência de equipamentos;

- Reusabilidade de código: se a aplicação e o seu código foram especificamente projetados, desenvolvidos e suportados para serem reutilizados em outras aplicações;
- Facilidade de instalação: qual foi o nível de preparação de procedimentos e ferramentas para instalação do sistema;
- Usabilidade: os procedimentos efetivos de inicialização, se backups e recuperação foram desenvolvidos e testados, a aplicação minimiza a necessidade de atividades manuais, tais como montagem de fitas magnéticas, manuseio de formulários e intervenção manual do operador;
- Portabilidade: se aplicação foi especificamente projetada, desenvolvida e suportada para ser instalada em múltiplas plataformas (Windows, Unix, Linux, etc);
- Facilidade de Manutenção - A aplicação foi especificamente projetada, desenvolvida para suportar manutenção, visando à facilidade de mudanças, por exemplo capacidade de consultas/relatórios flexíveis está disponível; dados de controle do negócio são agrupados em tabelas passíveis de manutenção pelo usuário;
- Concorrências: indica o volume de acesso simultâneo a aplicação;
- Características de segurança: qual o nível de segurança exigido pela aplicação;
- Acesso direto a dispositivos de terceiros: se há necessidade de se fazer considerações especiais no projeto do sistema para que a configuração do equipamento não fique sobrecarregada;
- Requer treinamento especial aos usuários: qual o nível de facilidade de treinamento de usuários.

A tabela abaixo contém os requisitos a serem preenchidos no EEDS, com seus respectivos pesos, e a relevância dada a cada um na aplicação do Módulo 02.

Tabela 06: Pesos dados aos Fatores Técnicos MOD02

Tfactor	Peso	Aplicação
Sistema distribuído	2	5- As funções de processamento são executadas dinamicamente na CPU mais apropriada

Desempenho da aplicação	1	4- Os requisitos de performance estabelecidos pelo usuário são rigorosos e requer tarefas de análise de performance na fase de análise do projeto
Eficiência do usuário final	1	4- A aplicação é considerada complexa, e possui requisitos estabelecidos para eficiência do usuário
Complexidade de processamento interno	1	4- Apresenta a maioria das categorias de processamento complexo
Reusabilidade do código	1	1- O código reutilizável é usado somente dentro da própria aplicação
Facilidade de instalação	0.5	5- Requisitos de instalação e ferramentas automatizadas foram fixados pelo usuário e roteiros de instalação foram preparados e testados
Usabilidade	0.5	1- Procedimentos eficientes de inicialização, backups e recuperação foram preparados, mas a intervenção do operador é necessária
Portabilidade	2	2- Necessidade de instalação em múltiplas plataformas foi levada em consideração, projeto foi desenvolvido para operar somente em ambientes similares de software ou hardware
Facilidade de manutenção	1	5- Dados de controle são mantidos em tabelas que podem ser utilizadas pelo usuário
Concorrências	1	3- Muitos acessos simultâneos foram fixados pelo usuário para a aplicação, o que força a execução de tarefas de análise de performance na fase de projeto da aplicação
Características de segurança	1	4- Um plano de segurança foi elaborado e testado para suportar o controle de acesso da aplicação
Acesso direto a dispositivos de terceiros	1	3- Necessidades especiais de processador para uma parte específica da aplicação
Requer treinamento especial aos usuários	1	4- Um plano de treinamento foi elaborado e testado para facilitar o uso da aplicação

Fonte: Elaboração própria

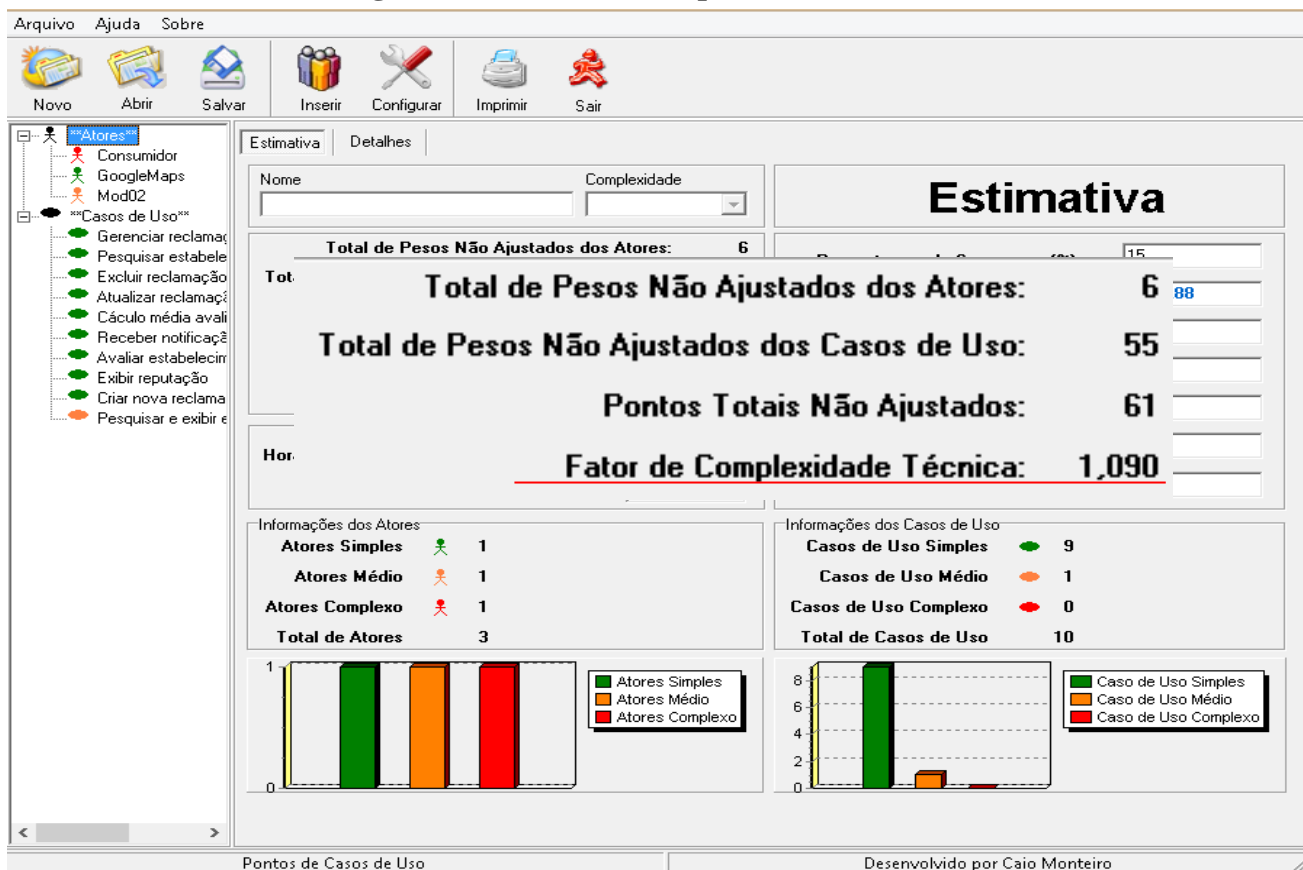
Após estabelecer os valores para cada requisito, foi feito o cálculo da complexidade dos fatores técnicos, chamado de TCF (Technical Complexity Factor) [3], a fórmula é dada por $TCF = 0.6 + (0.01 \times Tfactor)$. A tabela abaixo mostra como o resultado foi obtido pelo módulo 02, e a imagem subsequente contém a interface do EEDS, que realiza o cálculo automaticamente, com esse passo feito.

Tabela 07: Cálculo da Complexidade dos Fatores Técnicos MOD02

Tfactor	Peso	Aplicação	Resultado
Sistema distribuído	2	5	10
Desempenho da aplicação	1	4	4
Eficiência do usuário final	1	4	4
Complexidade de processamento interno	1	4	4
Reusabilidade do código	1	1	1
Facilidade de instalação	0.5	5	2.5
Usabilidade	0.5	1	0.5
Portabilidade	2	2	4
Facilidade de manutenção	1	5	5
Concorrências	1	3	3
Características de segurança	1	4	4
Acesso direto a dispositivos de terceiros	1	3	3
Requer treinamento especial aos usuários	1	4	4
TCF			1,090

Fonte: Elaboração própria

Figura 05 – Fator de Complexidade Técnica MOD02



Fonte: Elaboração própria

2.2.6 Fatores de Ajuste: Fatores Ambientais

Então foi feito o cálculo de fatores ambientais, também chamados de Efactor [3], que são aqueles que representam requisitos não-funcionais relacionados ao desenvolvimento sistema. Esses são:

- Familiaridade com o processo iterativo unificado: se os desenvolvedores estão familiarizados com esse processo;
- Experiência na aplicação: se os envolvidos no desenvolvimento possuem alguma experiência com a aplicação em desenvolvimento;
- Experiência em orientação a objetos: se os envolvidos têm algum tempo de experiência com programação orientada a objetos, que é a utilizada para o cálculo do PCU;
- Capacidade de liderança em análise: se os desenvolvedores estabelecem um objetivo e tem a capacidade de executá-lo;
- Motivação: se a equipe está motivada e inspirada a entregar um bom produto;

- Estabilidade de requisitos: se todos os requisitos definidos permanecem estáveis durante o desenvolvimento;
- Consultores part-time: se as horas de trabalho são distribuídas entre os membros;
- Dificuldade de programação na linguagem: quanto tempo de experiência os membros possuem em programação;

A tabela abaixo contém os requisitos a serem preenchidos no EEDS, com seus respectivos pesos, e a relevância dada a cada um na aplicação do Módulo 02.

Tabela 08: Pesos dados aos Fatores Ambientais MOD02

Efactor	Peso	Aplicação
Familiaridade com o processo interativo unificado	1,5	2- Um ou mais membros utilizou o processo uma ou poucas vezes
Experiência na aplicação	0,5	2- Todos os membros tem mais de 1.5 anos de experiência
Experiência em orientação a objetos	1	2- Todos os membros tem de 1 a 1.5 anos de experiência
Capacidade de liderança em análise	0,5	2- Possui experiência de poucos projetos
Motivação	1	3- A equipe está muito motivada para fazer um bom trabalho
Estabilidade de requisitos	2	2- Requisitos instáveis. Clientes demandam algumas mudanças realizadas em diversos intervalos
Consultores part-time	-1	1- Poucos membros (20%) trabalham em período parcial
Dificuldade de programação na linguagem	-1	2- Todos os membros tem mais de 1.5 anos de experiência

Fonte: Elaboração própria

Após estabelecer os valores para cada requisito, foi feito o cálculo da eficiência dos fatores ambientais, chamado de ECF (Environmental Complexity Factor) [3], a fórmula é dada por $ECF = 1.4 + (-0.03 \times Efactor)$. A tabela abaixo mostra como o resultado foi obtido pelo módulo 02, e a

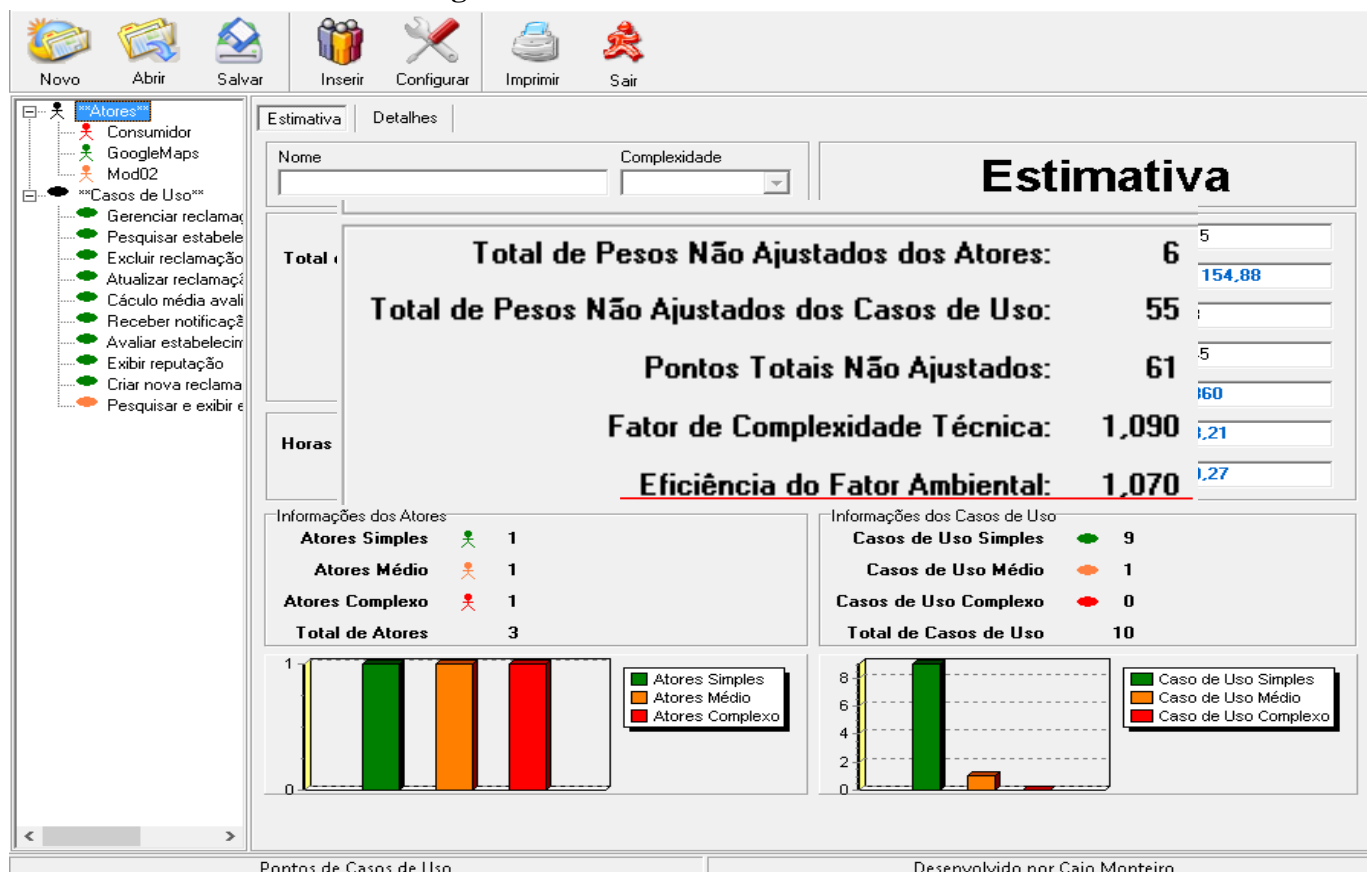
imagem subsequente contém a interface do EEDS, que realiza o cálculo automaticamente, com esse passo feito.

Tabela 09: Cálculo da Eficiência dos Fatores Ambientais MOD02

Efactor	Peso	Aplicação	Resultado
Familiaridade com o processo interativo unificado	1,5	2	3
Experiência na aplicação	0.5	2	1
Experiência em orientação a objetos	1	2	2
Capacidade de liderança em análise	0,5	2	1
Motivação	1	3	3
Estabilidade de requisitos	2	2	4
Consultores part-time	-1	1	0
Dificuldade de programação na linguagem	-1	2	1
ECF			1,070

Fonte: Elaboração própria

Figura 06 – Eficiência do Fator Ambiental MOD02



Fonte: Elaboração própria

2.2.7 Pontos de Caso de Uso

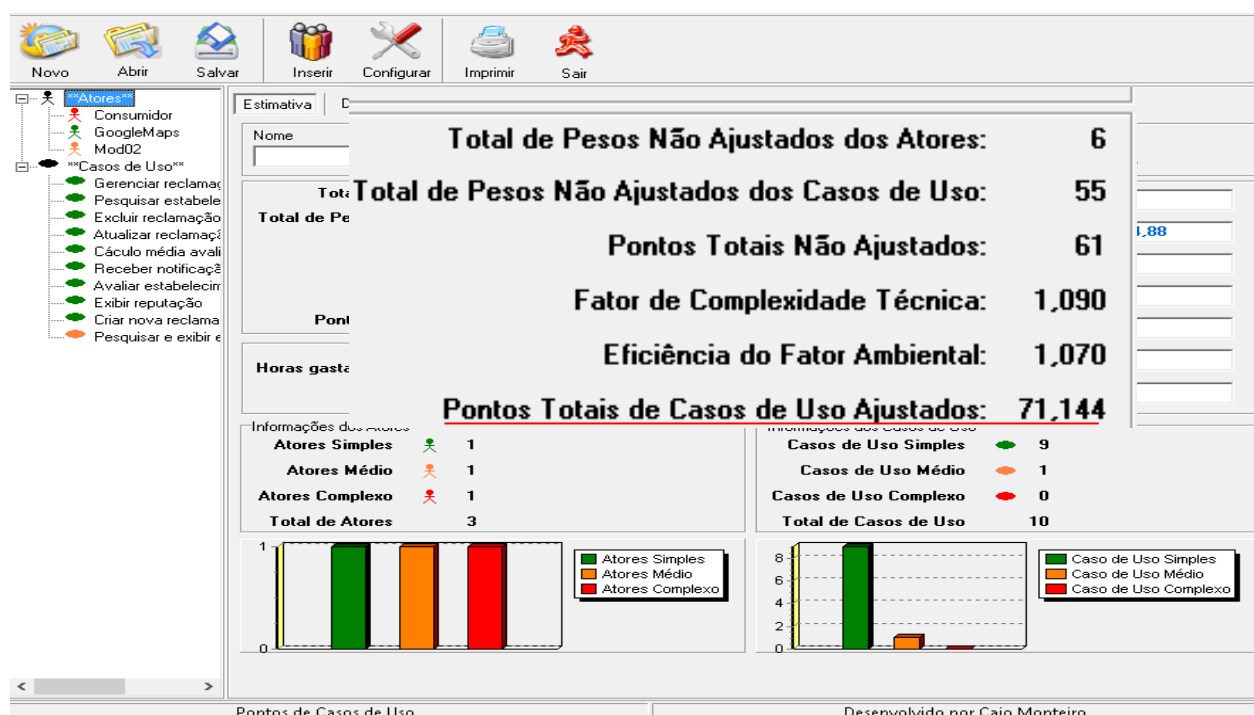
Por fim, foi calculado os Pontos de Caso de Uso multiplicando o UUCP pelo TCF e ECF obtidos. A tabela abaixo explica como foi realizado o cálculo e a imagem seguinte contém a interface do EEDS, que realiza o cálculo automaticamente, com isso feito.

Tabela 10: Cálculo dos Pontos de Caso de Uso MOD02

UUCP	TCF	ECF	PCU
61	1,090	1,070	71,144

Fonte: Elaboração própria

Figura 07 – Pontos de Caso de Uso MOD02



Fonte: Elaboração própria

2.2.8 Estimativa de Tempo de Trabalho

Estabelecidos os Pontos totais de Casos de Uso após o ajuste, foi realizado o cálculo estimado do tempo de trabalho, sendo que a média de horas por PCU pré-definido no EEDS foi de 36 horas por pessoa. Esse valor foi dado pela multiplicação de Pontos pela quantidade de horas por pessoa. A tabela abaixo mostra como foi feito o cálculo e a imagem seguinte contém a interface do EEDS, que realiza o cálculo automaticamente, com o resultado obtido.

Tabela 11: Cálculo do Tempo de Trabalho MOD02

PCU (total)	Horas gastas por PCU	Resultado
71,144	36	2561,18

Fonte: Elaboração própria

Figura 08 – Tempo de Trabalho MOD02

The screenshot shows a software application for estimating work time. The main window has a menu bar with icons for 'Novo', 'Abrir', 'Salvar', 'Inserir', 'Configurar', 'Imprimir', and 'Sair'. On the left, there is a sidebar with a tree view showing project elements like 'Consumidor', 'GoogleMaps', 'Mod02', and 'Casos de Uso'. The main area is divided into two tabs: 'Estimativa' and 'Detalhes'. The 'Estimativa' tab is active and displays a large text box with the text 'Horas gasta por pessoa para cada PCU.' and two input fields showing the values '36' and '2561,18'. Below this, there are two bar charts: 'Informações dos Atores' and 'Informações dos Casos de Uso'. The 'Atores' chart shows 1 Simple, 1 Medium, and 1 Complex actor, totaling 3. The 'Casos de Uso' chart shows 9 Simple, 1 Medium, and 0 Complex cases, totaling 10. The bottom status bar indicates 'Pontos de Casos de Uso' and 'Desenvolvido por Caio Monteiro'.

Fonte: Elaboração própria

2.2.9 Custo de Desenvolvimento

Ao estabelecer a quantidade de horas trabalhadas no projeto Reclame São João pelo módulo de Reclamações, foi feita a estimativa de Custo de Desenvolvimento do sistema, esse valor varia em função do papel de cada membro da equipe, ou seja, de sua especialização na tarefa. O valor estabelecido pelos estudantes responsáveis pelo projeto por hora de desenvolvimento de cada Caso de Uso foi de 11,50 reais [9]. Então, manualmente, multiplicou-se esse valor médio pelo número de Pontos de Caso de Uso (com a margem de segurança proposta pelo EEDS). A tabela abaixo mostra como foi obtido o resultado e qual foi o custo de desenvolvimento do Módulo 02.

Tabela 12: Cálculo do Custo de Desenvolvimento MOD02

Valor médio da hora de trabalho	UCP + Margem de segurança	Custo de desenvolvimento
11,50	2.945,36	33.871,64

Fonte: Elaboração própria

2.2.10 Estimativas de Tempo

O software calcula a estimativa da dedicação mensal da equipe, do total de meses para o desenvolvimento do sistema e a conversão em anos. Para esse cálculo foi necessário que se inserisse a quantidade de membros na equipe e a dedicação mensal de cada membro (em horas), então o programa fez a multiplicação.

O módulo 02 possui oito integrantes, sendo três desenvolvedores, dois desenvolvedores de banco de dados e três analistas e testadores. Cada um deles trabalha no projeto por aproximadamente 45 horas por mês, considerando uma média de 4 semanas, com 21 horas de aulas de Projeto de Desenvolvimento de Sistema, e 14 horas de aulas de Aplicações para Web 2; e 10 horas de dedicação em horários extracurriculares. A tabela abaixo contém a aplicação no módulo de Reclamações e a imagem subsequente contém a interface do EEDS, que realiza o cálculo automaticamente e o resultado da dedicação mensal da equipe obtido.

Tabela 13: Cálculo da Dedicação Mensal MOD02

Quantidade de membros	Dedicação mensal de cada um	Dedicação mensal da equipe
8	45	360

Fonte: Elaboração própria

Figura 09 – Dedicação Mensal MOD02

The screenshot shows the EEDS software interface. On the left, there is a tree view under 'Casos de Uso' with the following items: Gerenciar reclamações, Pesquisar estabelecimento no google maps, Excluir reclamação, Atualizar reclamação, Cálculo média avaliação estabelecimento, Receber notificação resposta reclamação, Avaliar estabelecimento, Exibir reputação, Criar nova reclamação, and Pesquisar e exibir estabelecimento. The main area has a tabbed interface with 'Estimativa' selected. It contains input fields for 'Porcentagem de Segurança (%)' (15), 'UCP + Margem de Segurança' (2945,36), 'Quantidade de Membros na Equipe' (8), 'Dedicação Mensal de cada membro' (45), 'Dedicação Mensal da Equipe' (360), 'Total de Meses' (8,18), and 'Total de Anos' (0,68). A summary table on the right shows the calculated values for each input. At the bottom, there are two bar charts and a legend for 'iso de Uso Simples', 'iso de Uso Médio', and 'iso de Uso Complexo'.

Nome	Complexidade	Estimativa
Porcentagem de Segurança (%)	15	15
UCP + Margem de Segurança	2945,36	2945,36
Quantidade de Membros na Equipe	8	8
Dedicação Mensal de cada membro	45	45
Dedicação Mensal da Equipe	360	360
Total de Meses	8,18	8,18
Total de Anos	0,68	0,68

iso de Uso Simples
iso de Uso Médio
iso de Uso Complexo

Fonte: Elaboração própria

Com base na dedicação mensal resultante, o EEDS fez a estimativa do total de meses necessários para a conclusão do projeto, dividindo o total de pontos de caso de uso (incluindo a margem de segurança) pela dedicação mensal obtida da equipe. A tabela abaixo contém a aplicação no módulo de Reclamações e a imagem subsequente contém a interface do EEDS, que realiza o cálculo automaticamente e o resultado do total de meses obtido.

Tabela 14: Cálculo do Tempo de Execução em meses MOD02

UCP+Margem de segurança	Dedicação mensal da equipe	Total de meses
2945,36	360	8,18

Fonte: Elaboração própria

Figura 10 – Estimativa de Tempo de Execução em meses MOD02

The screenshot shows the EEDS software interface for estimating project execution time. The 'Estimativa' tab is active, displaying various input fields and calculated results. The 'Total de Meses' is calculated as 8,18, and the 'Total de Anos' is 0,68. The interface also includes a tree view of use cases on the left and two bar charts at the bottom.

Fonte: Elaboração própria

Então com uma fórmula simples de conversão (o produto da divisão do total de meses pela quantidade de meses no ano), o programa calculou o total de anos correspondente. A tabela abaixo

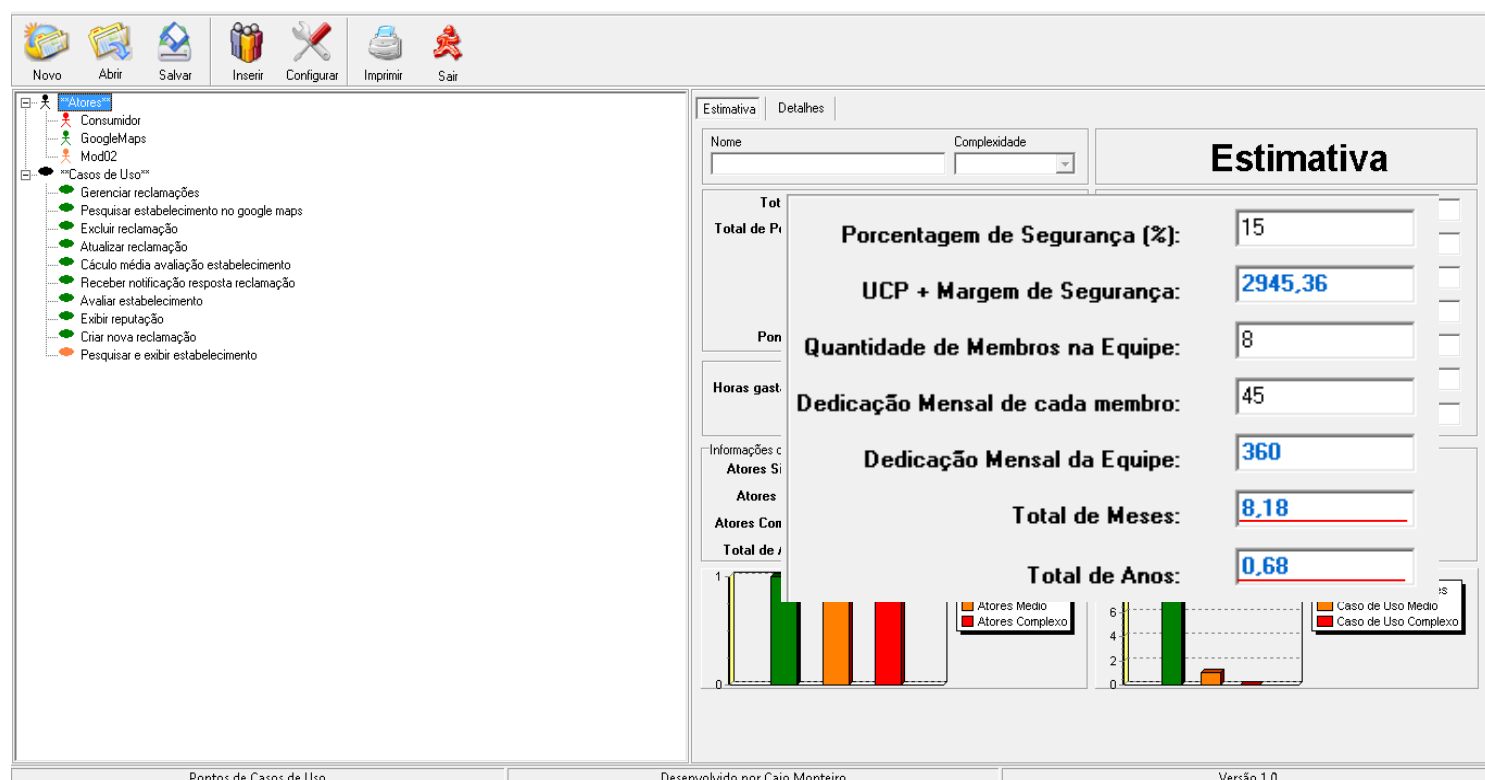
contém a aplicação no módulo de Reclamações e a imagem subsequente contém a interface do EEDS, que realiza o cálculo automaticamente e o resultado do total de anos obtido.

Tabela 15: Cálculo do Tempo de Execução em anos MOD02

Total de meses	Total de anos
8,18	0,68

Fonte: Elaboração própria

Figura 11 – Estimativa de Tempo de Execução em anos MOD02



Fonte: Elaboração própria

2.2.11 Estimativa do Projeto Reclame São João

O site completo foi dividido em cinco módulos, sendo eles o MOD01: Usuários, encarregado pelos perfis de usuário no site; o MOD02: Reclamações, responsável pela gestão de reclamações feitas pelos clientes; o MOD03: Estabelecimentos, sobre a gestão de estabelecimentos e suas ações; o MOD04: Relatórios, encarregado pela gestão de relatórios tabulares e gráficos; e o MOD05: Administrativo, responsável pela gestão dos administradores e suas ações [9]. Cada módulo realizou a estimativa de esforços da mesma maneira feita pelo de Reclamações, cada qual

com seus Casos de Uso e variações nos Fatores Técnicos e Ambientais; obtendo resultados um pouco diferentes, que foram consultados para a pesquisa e somados, para confirmar se o orçamento feito na abertura do projeto estava correto. A tabela abaixo mostra o custo de desenvolvimento obtido por cada módulo, e o custo do projeto como um todo após ser feita a soma dos módulos.

Tabela 16: Cálculo do Custo do Projeto Reclame São João

Módulos	Custo de desenvolvimento
Mod01: Usuários	58.178,84
Mod02: Reclamações	33.871,64
Mod03: Estabelecimentos	23.470,35
Mod04: Relatórios	24.146,43
Mod05: Administrativo	37.573,26
Custo do Projeto	177.240,52

Fonte: Elaboração própria

Considerando que no início do ano, no Termo de Abertura do Reclame São João, o orçamento foi dado por 218.926,66 reais, baseando-se no piso salarial de um técnico de informática júnior trabalhando em empresa de médio porte, mais um acréscimo de 20% do valor contabilizando o transporte e energia utilizada e a inflação [9]; é notável a diferença de valores da estimativa baseada em todos os cálculos efetuados por todos os módulos da mesma maneira feita nesse trabalho, do orçamento feito sem visão do produto e sem a base técnica necessária. A tabela abaixo mostra a variação entre os dois valores.

Tabela 17: Cálculo entre Diferença do Custo Inicial e Custo utilizando o EEDS do Projeto

Orçamento inicial	Orçamento pelo esforço	Diferença
218.926,66	177.240,52	41.686,14

Fonte: Elaboração própria

Conclusões e Recomendações

Utilizar um programa que meça os esforços num software para valorizá-lo pode ser muito útil para o desenvolvedor dele, pois estabelecer um preço de um grande projeto sem ter experiência, fundamentando-se em fatores como piso salarial de um profissional, ou gasto de energia pode trazer equívocos. Em tal grau que o resultado da estimativa obtido na pesquisa foi superior que o obtido na fase de análise do projeto, pois no início não havia muita experiência com o cálculo, mas após toda a pesquisa realizada para o trabalho, houve melhor compreensão. E através da realização é possível chegar a uma estimativa mais realista, porque se adquire uma noção mais ampla do produto, e ao decorrer do processo ocorrem muitas mudanças.

Quando se faz o orçamento sem embasamento, dificilmente se leva em consideração condições estudadas nesse trabalho, por exemplo, o nível de experiência dos envolvidos com aquele tipo de projeto. Pouca experiência não desvaloriza o sistema. Pelo contrário, o valoriza, porque o esforço necessário para desenvolvê-lo é maior. A ferramenta EEDS-PCU (Estimativa de Esforço de Desenvolvimento de Software com Ponto de Casos de Uso), de Caio Monteiro Barbosa da Silva contém todos os fatores que devem ser considerados, e o desenvolvedor apenas precisa saber identificar os números referentes ao seu projeto de forma mais realista possível para obter resultados precisos.

Foi concluído que utilizar a estimativa de esforços pelos Pontos de Caso de Uso é um método muito recomendado por ser justo, e que a EEDS facilita o cálculo para quem não possui experiência com a técnica. Por ser uma ferramenta não atualizada desde sua criação em 2006, recomenda-se que sejam realizadas novas pesquisas a respeito da estimativa de esforços por Pontos de Caso de Uso, buscando estudar novos softwares e possíveis adaptações do método, pois a cada ano que se passa a tecnologia evolui, e o mercado digital exige acompanhamento.

3 Referências Bibliográficas

- [1] Fiorini, S. T., Staa, A., Baptista, R. M. (1998) Engenharia de Software com CMM, Brasport.
- [2] Vazquez, C.E., Simões, G.S., Albert, R.M. (2005) Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software, Editora Érica, 3ª edição.
- [3] KARNER, G. Use Case Points: resource estimation for Objectory projects. Objective Systems SF AB (copyright owned by Rational/IBM), 1993
- [4] Scheinder, G. e Winters, J. P. (2001) Applying Use Cases: A Pratical Guide, 2nd Edition
- [5] Rumbaugh, J. e Jacobson, I., The Unified Modeling Language User Guide, Addison Wesley Longman, 1999
- [6] Valente, F. F. R. e Falbo, R. A. (2002) Uso de Gerência de Conhecimento para Apoiar a Realização de Estimativas, In: Proceedings of the XXVIII Latin-American Conference on Informatics - CLEI'2002, Montevideo, Uruguay, November 2002.
- [7] Pressman, R. S. (2005) Software Engineering: A Practitioner's Approach, Mc Graw Hill, 6th edition.
- [8] Gonçalves, B. P, Souza, E. G., Nogueira, G.P, Dattolli, R.P, Freitas, A., Cazaroto, G., Costa, G.P, Araújo, L.A. (2017) Módulo 02 – Reclamações, Modelo de Casos de Uso, V.2.3
- [9] Termo de Abertura do Projeto Reclame São João, 2017