

**UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI**  
**Bacharelado em Sistemas de Informação**  
**Gabriel Henrique da Costa Rufino**

**MÉTRICAS DE ESTIMATIVA DE SOFTWARE APLICADAS AO  
DESENVOLVIMENTO DE UM SISTEMA DE PROTOCOLOS**

**Diamantina**  
**2018**



**Gabriel Henrique da Costa Rufino**

**MÉTRICAS DE ESTIMATIVA DE SOFTWARE APLICADAS AO  
DESENVOLVIMENTO DE UM SISTEMA DE PROTOCOLOS**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Bacharelado em Sistemas de  
Informação, como parte dos requisitos exigidos  
para a conclusão do curso.

Orientador: Marcelo Ferreira Rego

**Diamantina  
2018**




**Gabriel Henrique da Costa Rufino**

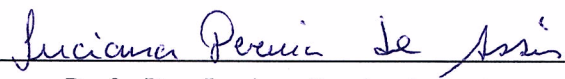
**MÉTRICAS DE ESTIMATIVA DE SOFTWARE APLICADAS AO  
DESENVOLVIMENTO DE UM SISTEMA DE PROTOCOLOS**

Trabalho de Conclusão de Curso apresentado  
ao Curso de Bacharelado em Sistemas de  
Informação, como parte dos requisitos exigidos  
para a conclusão do curso.

Orientador: Prof. Me. Marcelo Ferreira Rego

Data de aprovação 02/03/18

  
\_\_\_\_\_  
Prof. Me. Marcelo Ferreira Rego  
Departamento de Computação - UFVJM

  
\_\_\_\_\_  
Profa. Dra. Luciana Pereira de Assis  
Departamento de Computação - UFVJM

  
\_\_\_\_\_  
Profa. Dra. Maria Lúcia Bento Villela  
Departamento de Computação - UFVJM

Diamantina  
2018



A Deus e minha família, pelo apoio e amor incondicional.





## **AGRADECIMENTOS**

Gostaria de dedicar este trabalho primeiramente a Deus pelas oportunidades que colocou em minha vida. Aos meus pais pelo apoio, amor e suporte incondicional. À minha irmã, pelo carinho e amor sempre.

Agradeço ao meu orientador Marcelo Ferreira Rego pela orientação, amizade e apoio durante a realização do trabalho. Aos outros professores do curso de Sistemas de Informação da UFVJM pelo conhecimento passado a mim, que foi imprescindível para a realização deste.

Finalmente gostaria de agradecer aos meus amigos de Diamantina pelo carinho e apoio em todos os momentos.



## RESUMO

Estudos envolvendo processos de desenvolvimento de software têm se voltado para uma busca pela otimização, melhor qualidade e menores custos. As métricas de software são técnicas que têm como objetivo auxiliar no aperfeiçoamento do desenvolvimento de software, por meio de estimativas. Este trabalho tem como meta a análise da aplicação de duas das métricas de software, Planning Poker e Análise de Pontos de Função, usados para estimar o esforço e prioridade no desenvolvimento de um Sistema de Protocolos. Além da análise e comparação das duas métricas, sugeriu-se uma metodologia para personalizar o uso da Análise de Pontos de Função, utilizando Intervalo de Confiança. Os resultados do estudo mostraram que a Análise de Pontos de Função surtiu bons resultados, tendo como parâmetro uma fonte de conversão de pontos de função em horas implementada de acordo com a realidade do projeto, utilizando Intervalo de Confiança. A técnica de Planning Poker mostrou-se muito pertinente à utilização paralela com a metodologia ágil de desenvolvimento, integrando melhor a equipe do projeto e resultando em estimativas pontuais.

Palavras-chave: Estimativas. Planning Poker. Análise de Pontos de Função. Métricas de Software. Intervalo de Confiança.



## **ABSTRACT**

The studies involving software development processes have been focused on a search for the optimization of these processes, seeking better quality and lower costs. Software metrics are techniques that aim to aid in the improvement of software development, by means of estimates. The purpose of this work is to analyze the application of two software metrics, Planning Poker and Function Point Analysis, used to estimate the effort and priority in the development of a Protocol System. In addition to the analysis and comparison of the two metrics, a methodology was suggested to customize the use of Function Point Analysis, using statistical techniques. The results of the study showed that the Functional Point Analysis had good results, having as parameter a source of conversion of function points in hours implemented according to the reality of the project, using Confidence Interval. The Planning Poker technique was very pertinent to the parallel use with the agile development methodology, better integrating the project team and resulting in punctual estimates.

**Keywords:** Estimate. Planning Poker. Function Point Analysis. Software Metrics. Confidence Interval.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Distribuição T Student. . . . .	43
Figura 2 – Resultados teste de normalidade. . . . .	52
Figura 3 – Resultados teste de normalidade. . . . .	52
Figura 4 – Gráfico de Normalidade. . . . .	53
Figura 5 – Diagrama Entidade-Relacionamento do SIP. . . . .	61
Figura 6 – Documentação de Tarefas. . . . .	62
Figura 7 – Tela página inicial do SIP. . . . .	63
Figura 8 – Tela cadastro de protocolo do SIP. . . . .	63
Figura 9 – Tela lista de protocolos do SIP. . . . .	64





## LISTA DE TABELAS

Tabela 1 – Tabela de complexidade funcional dos ALI e AIE. . . . .	38
Tabela 2 – Tabela de complexidade para entradas externas (EEs). . . . .	38
Tabela 3 – Tabela de complexidade para saídas externas (SEs) e consultas externas (CEs). . . . .	38
Tabela 4 – Tabela de contribuição dos pontos de função não ajustados das funções do tipo dado. . . . .	38
Tabela 5 – Tabela de contribuição dos pontos de função das funções do tipo transação. . . . .	39
Tabela 6 – Tabela de Produtividade Mínima. . . . .	41
Tabela 7 – Resultados da aplicação da APF. . . . .	45
Tabela 8 – Resultados da aplicação da APF. . . . .	46
Tabela 9 – Tabela de Contribuição das Funções do Tipo Dado e do Tipo Transação. . . . .	47
Tabela 10 – Tabela de Aplicação do Fator de Ajuste. . . . .	49
Tabela 11 – Amostra de tempo gasto para desenvolvimento de algumas funcionalidades do SIP. . . . .	51
Tabela 12 – Histórias de Usuário e respectivas pontuações coletadas na primeira reunião. . . . .	54
Tabela 13 – Resultados da estimativa de esforço para tarefas na segunda reunião. . . . .	55
Tabela 14 – Histórias de Usuário e respectivas pontuações para prioridade coletadas na segunda reunião. . . . .	56
Tabela 15 – Resultados obtidos. . . . .	57



## **LISTA DE ABREVIATURAS E SIGLAS**

UFVJM - Universidade Federal dos Vales do Jequitinhonha e Mucuri

ANCINE - Agência Nacional do Cinema.

IFPUG - International Function Point Users Group.

HTML - HyperText Markup Language.

PHP - Hypertext Preprocessor.

SQL - Structured Query Language.

CSS - Cascading Style Sheets.

IBM - International Business Machines.



## LISTA DE SÍMBOLOS

$\alpha$  Alfa



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> . . . . .	<b>23</b>
1.1	Objetivo . . . . .	24
1.2	Justificativa . . . . .	24
1.3	Organização . . . . .	25
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> . . . . .	<b>27</b>
2.1	Scrum . . . . .	27
2.1.1	Visão Geral . . . . .	27
2.1.2	Times e Papéis . . . . .	27
2.1.3	Artefatos . . . . .	27
2.1.4	Eventos . . . . .	27
2.2	Métricas de Software . . . . .	28
2.2.1	Planning Poker . . . . .	28
2.2.2	Análise de Pontos de Função . . . . .	28
<b>3</b>	<b>REVISÃO DA LITERATURA</b> . . . . .	<b>31</b>
3.1	Scrum . . . . .	31
3.2	Análise de Pontos de Função . . . . .	31
3.3	Planning Poker . . . . .	32
<b>4</b>	<b>METODOLOGIA</b> . . . . .	<b>33</b>
4.1	Processo de Desenvolvimento com SCRUM . . . . .	33
4.2	Estimativa de Software . . . . .	34
4.2.1	Planning Poker . . . . .	34
4.2.1.1	Estimativa de Esforço . . . . .	34
4.2.1.2	Estimativa de Prioridade . . . . .	35
4.2.2	Análise de Pontos de Função . . . . .	35
4.3	Intervalo de Confiança . . . . .	41
<b>5</b>	<b>RESULTADOS</b> . . . . .	<b>45</b>
5.1	Análise de Pontos de Função . . . . .	45
5.1.1	Aplicando Intervalo de Confiança à Análise de Pontos de Função . . . . .	50
5.2	Planning Poker . . . . .	54
<b>6</b>	<b>DISCUSSÃO</b> . . . . .	<b>57</b>
6.1	Análise de Pontos de Função . . . . .	57
6.2	Planning Poker . . . . .	58

<b>7</b>	<b>O SISTEMA</b> . . . . .	<b>61</b>
<b>8</b>	<b>CONCLUSÃO</b> . . . . .	<b>65</b>
<b>9</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>67</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>69</b>



## 1 INTRODUÇÃO

As estimativas aplicadas ao desenvolvimento de software referentes a esforço e custos não têm se mostrado eficientes na geração de dados precisos, sendo provado em artigos, estudos de caso e pesquisas. (HAUGEN, 2006). Não existem muitos estudos sobre a combinação de estimativas em desenvolvimento de software. Entretanto, alguns indícios mostram que a estimação tende a reduzir o otimismo, isto é, diminuição de equívocos com relação ao processo de desenvolvimento. Estimativas têm por objetivo organizar o processo de desenvolvimento, priorizando requisitos que são mais importantes para o sistema. (HAUGEN, 2006).

As estimativas podem ser estruturadas e não-estruturadas. De acordo com Haugen (2006), estimativas estruturadas exigem conhecimento e técnicas previamente selecionadas e analisadas. Já as estimativas não-estruturadas são informais, com reuniões sem roteiros, gerando um consenso no final. Técnicas não-estruturadas de estimativas não têm surtido os efeitos esperados, pelo fato de que nas reuniões, onde participam os desenvolvedores, clientes e analistas de negócios, não existe um acordo com relação a ordem dos requisitos a serem implementados. Requisito é “uma condição, característica ou capacidade, determinada no universo das necessidades do negócio do usuário, que deve ser atendida por um sistema na forma de aspectos funcionais ou não funcionais”. (VAZQUEZ; SIMOES; ALBERT, 2010, p. 43). Desenvolvedores, geralmente, preferem implementar primeiro os requisitos de alto risco, enquanto que os clientes preferem que sejam implementados requisitos de maior prioridade para o sistema. Além disso, nas reuniões envolvendo métricas não-estruturadas, percebe-se que nem todos os envolvidos no projeto têm suas opiniões ouvidas. (HAUGEN, 2006).

Segundo Vazquez, Simoes e Albert (2010), os requisitos do software são formalmente apresentados pelo cliente antes do processo de desenvolvimento iniciar-se. Entretanto, é muito comum que novos requisitos surjam durante o processo de desenvolvimento, alterando a estimativa inicial para os recursos que seriam utilizados (caso exista alguma estimativa). Ainda de acordo com Vazquez, Simoes e Albert (2010), a medição de software oferece mecanismos para que, durante todo o processo de desenvolvimento, as estimativas para a utilização de recursos seja a mais precisa possível. Um dos objetivos das métricas de software é permitir que os problemas sejam identificados e corrigidos antes que prejudiquem o andamento do projeto. (VAZQUEZ; SIMOES; ALBERT, 2010). As principais métricas de estimativa de software encontradas na literatura são: Análise de Pontos de Função (APF), Planning Poker (PP), Método de Delphi, Contagem por linhas de código e Ideal Day.

O Método de Delphi não se baseia em encontros presenciais, mas sim em estimativas realizadas à distância e individuais, sempre tendo um moderador como supervisor. É um método iterativo até que se chegue a um consenso entre as partes envolvidas. Apesar de ser um método conhecido, não há registros que apontem vantagens em relação às métricas estruturadas. O Método de Delphi é uma métrica estruturada de estimativa. (MOLOKKEN-OSTVOLD;

[HAUGEN, 2007](#)).

A Contagem por linhas de código é um método simples, porém controverso. Os que o utilizam não chegam a um consenso com relação a contagem das linhas, sendo que muitos alegam que os comentários e linhas em branco não deveriam ser contabilizados. Outro limitante seria a linguagem de programação, pois esta só pode servir de documentação para projetos que utilizem a mesma linguagem. A técnica é uma métrica estruturada. ([CORDEIRO, 2018](#)).

Ideal Day é uma métrica que se baseia na quantidade de trabalho realizado por um desenvolvedor em um dia, sendo assim, as tarefas são definidas por dia de trabalho. Apesar de apresentar bons resultados, os prazos diários poderão ser alterados no decorrer do projeto devido a trocas de membros na equipe ou outros problemas não planejados. A técnica é uma métrica estruturada. ([GAMBA; BARBOSA, 2010](#)).

Dentre as métricas analisadas, foram escolhidas duas para compor o estudo, o Planning Poker e a Análise de Pontos de Função, sendo que, segundo as definições de [Haugen \(2006\)](#), as duas são métricas estruturadas. As mesmas foram escolhidas pelo seu uso difundido, tanto no meio da indústria quanto no acadêmico ([HAUGEN, 2006](#)). Planning Poker é utilizado para análise de estimativas em grupo, ([HAUGEN, 2006](#)), em desenvolvimento ágil dentro de empresas, ([DIMITRIJEVIC; JOVANOVIC; DEVEDZIC, 2014](#)) e em estimativas, planejamento e análise de projetos Web. ([TORRECILLA-SALINAS et al., 2015](#)). Já a Análise de Pontos de Função possui um histórico de utilização em empresas como IBM, para medir produtividade de projetos de software e também em contratos públicos de desenvolvimento e manutenção de software. ([VAZQUEZ; SIMOES; ALBERT, 2010](#)).

O software escolhido para a aplicação das métricas é o Sistema de Protocolos (SIP), idealizado para otimizar o gerenciamento dos documentos que tramitam na Pró-Reitoria de Graduação (PROGRAD) da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM), por meio de uma plataforma Web.

## 1.1 Objetivo

O objetivo do estudo é aplicar as métricas Análise de Pontos de Função e Planning Poker no contexto do desenvolvimento do SIP e, posteriormente, comparar os resultados. A métrica APF realiza estimativa de esforço em horas. Enquanto que a métrica PP faz estimativa de esforço e também de prioridade relativas ao desenvolvimento de software.

## 1.2 Justificativa

As estimativas aplicadas ao desenvolvimento de software ainda são processos que exigem técnicas de medição mais acuradas, a fim de gerar resultados mais precisos. Além disso, a indústria de software é dinâmica, sendo que constantes melhorias no processo de desenvolvimento são exigidas para que a qualidade desejada seja alcançada. ([HAUGEN, 2006](#)). Sendo assim, este trabalho aplica e analisa duas métricas de software, PP e APF, para que

sejam determinadas as melhores práticas para cada uma delas, a fim de otimizar a estimação de software. Ambas as métricas estão inseridas no contexto da metodologia ágil baseado no Scrum.

### **1.3 Organização**

Para a realização deste trabalho foi feito um estudo sobre as duas métricas de software e seu histórico de aplicação, disposto no capítulo 2. Um breve histórico sobre a literatura já estudada é apresentado no capítulo 3. Definiu-se uma metodologia de trabalho para a aplicação das duas métricas no capítulo 4. Foram apresentados os resultados do estudo e das aplicações das métricas no capítulo 5. O capítulo 6 analisa os resultados da aplicação de cada métrica. O capítulo 7 apresenta os detalhes sobre o sistema utilizado para a aplicação das métricas. No capítulo 8, são apresentadas as conclusões do trabalho. Finalmente, o capítulo 9 apresenta as considerações finais.



## 2 REFERENCIAL TEÓRICO

As métricas de software são ferramentas ou metodologias que têm como objetivo auxiliar no desenvolvimento de sistemas computacionais, realizando estimativas de algumas variáveis relacionadas ao sistema. (GAMBA; BARBOSA, 2010). As variáveis consideradas nas principais métricas são: custos, esforço e tempo. Como as empresas precisam aliar excelência a baixo custo, o uso de métricas de software é uma boa solução para prever a quantidade de esforço que deve ser empregada ao desenvolvimento de um sistema, permitindo assim a redução dos riscos e o aumento da qualidade deste processo. (GAMBA; BARBOSA, 2010).

### 2.1 Scrum

#### 2.1.1 Visão Geral

De acordo com Schwaber e Sutherland (2016), o Scrum é um *framework* que possibilita o tratamento de problemas complexos. O mesmo pode ser composto por times, papéis, artefatos e eventos.

#### 2.1.2 Times e Papéis

Segundo Gamba e Barbosa (2010), *Scrum Master* é aquele que gerencia e lidera a equipe de desenvolvimento. O uso da metodologia Scrum exige ainda, a definição do *Product Owner*, que é a pessoa responsável por definir quais características o sistema deve apresentar. (GAMBA; BARBOSA, 2010). Ainda segundo Gamba e Barbosa (2010), *Stakeholder* é a pessoa que possui algum interesse no desenvolvimento do sistema.

#### 2.1.3 Artefatos

Segundo Primo (2011), histórias de usuários “são artefatos utilizados no desenvolvimento de sistemas que são geridos segundo metodologias ágeis”. Elas têm por objetivo retratar as necessidades do cliente, e, portanto, o próprio cliente é o responsável pela sua elaboração.

#### 2.1.4 Eventos

De acordo com Gamba e Barbosa (2010), o Scrum possui alguns eventos ou regras a serem definidos, de acordo com a necessidade de cada projeto. Geralmente são realizados encontros para definir o conjunto de requisitos do software. O conjunto de requisitos que ainda não foi implementado é chamado de *Backlog*. Cada iteração do processo de desenvolvimento é chamado de *Sprint*, sendo que nela um conjunto de requisitos presentes no *Backlog* é implementado. Além disso, cada *Sprint* é realizada em intervalos de tempo, geralmente de um mês. No início de cada *Sprint* é realizada a *Planning Meeting*, que define uma lista de prioridades para a

implementação dos requisitos, denominada *Sprint Backlog*. A *Sprint Review Meeting* é realizada sempre no último dia da *Sprint* para avaliar o que foi feito.

## 2.2 Métricas de Software

As métricas de software são aplicadas com o intuito de coletar e analisar os dados de medição. (GAMBA; BARBOSA, 2010). De acordo com Haugen (2006), existe uma grande necessidade da aplicação de métricas nos projetos de software, uma vez que elas podem favorecer na conquista de patrocínios, na qualidade do planejamento e na alocação de recursos materiais e humanos.

### 2.2.1 Planning Poker

Planning Poker é realizada por meio de reuniões, onde cada participante recebe um número igual de cartas com números, sendo que as mesmas são utilizadas para inferências com relação ao esforço ou prioridade necessários para o desenvolvimento de certa funcionalidade do sistema. As funcionalidades podem estar representadas na forma de história de usuário ou tarefas. A cada história de usuário ou tarefa elucidada, cada participante mostra uma carta com o número que acredita ser referente ao esforço ou prioridade para desenvolvê-las. Além da técnica ser importante para o controle da qualidade dos processos, a mesma auxilia na interação da equipe de desenvolvimento, influenciando no bom andamento do projeto. (HAUGEN, 2006).

Segundo Haugen (2006), Planning Poker é uma técnica em que a equipe de desenvolvimento utiliza um conjunto de cartas, onde os valores impressos nas mesmas variam de acordo com a escala desejada, podendo esta ser linear ou até seguindo a sequência Fibonacci, a fim de estimar o nível de esforço necessário para desenvolver cada requisito.

De acordo com Gamba e Barbosa (2010), Planning Poker foi introduzida por James Grenning em 2002 como sendo uma técnica baseada em discussões em grupo, onde os membros da equipe de desenvolvimento, juntamente com o *Product Owner* e *Stakeholders* utilizam cartas de baralho para estimar valores de esforço (em horas) ou prioridade, que devem ser aplicados para cada história de usuário.

### 2.2.2 Análise de Pontos de Função

A Análise de Pontos de Função (APF) é uma metodologia que emprega a medição de software por meio de processos técnicos que classificam as funções do sistema de acordo com suas complexidades. (VAZQUEZ; SIMOES; ALBERT, 2010). De acordo com Lopes e Braga (2011), a técnica surgiu na década de 70 na IBM como uma alternativa às métricas baseadas em linhas de código. O objetivo da métrica é oferecer ao desenvolvedor uma visão técnica do sistema, em relação ao tamanho funcional do mesmo, proporcionando um maior controle de qualidade e de custos durante a fase de desenvolvimento. Das métricas de software, a APF

é uma das mais precisas no que diz respeito às técnicas de medição, oferecendo segurança e confiabilidade ao desenvolvedor. (VAZQUEZ; SIMOES; ALBERT, 2010).

“A análise de pontos de função é um método padrão para medir o desenvolvimento de software do ponto de vista do usuário”. (VAZQUEZ; SIMOES; ALBERT, 2010, p. 43). A mesma avalia e atribui um tamanho funcional ao software, isto é, a quantidade de funções que o código conterá. (DEKKERS, 1999). É uma técnica que tradicionalmente é composta de vários processos que envolvem tipo de contagem, definição do escopo de avaliação e a delimitação das fronteiras do software. (VAZQUEZ; SIMOES; ALBERT, 2010). Segundo Lopes e Braga (2011), a APF pode ser adaptada para gerar resultados em forma de esforço, em horas.





### 3 REVISÃO DA LITERATURA

Com o propósito de enriquecer o estudo, artigos e trabalhos foram reunidos e analisados, com o intuito de entender como a metodologia desenvolvida neste trabalho já havia sido estudada e quais foram as conclusões obtidas por outros autores.

#### 3.1 Scrum

De acordo com [Bissi \(2007\)](#), a utilização do Scrum oferece diversos benefícios como redução de riscos, maior integração entre membros de uma mesma equipe, solução rápida para problemas, inclusão dos clientes no processo de desenvolvimento, entre outros.

[Gamba e Barbosa \(2010\)](#) aplicaram as métricas em um ambiente ágil de desenvolvimento e concluíram que a aplicação do Scrum aliado à métricas de software como Análise de Pontos de Função e Planning Poker aumenta a qualidade das estimativas.

[Schwaber e Sutherland \(2016\)](#) destacam que o Scrum pode ser aplicado em partes dentro de projetos, no entanto, os artefatos, papéis, eventos e regras que são aplicados ao projeto devem ser imutáveis.

#### 3.2 Análise de Pontos de Função

Segundo [Lopes e Braga \(2011\)](#), após analisar diversas técnicas de estimação como contagem por linha de código, contagem de telas e outras, e compará-las à Análise de Pontos de Função, a última se destacou, gerando resultados ótimos e satisfazendo às necessidades do mercado.

Uma revisão da literatura foi realizada por [Junior, Fantinato e Sun \(2015\)](#), com o objetivo de verificar a precisão do uso de Análise de Pontos de Função para medir o tamanho funcional de software. As primeiras conclusões do estudo demonstraram que a técnica é muito madura e consistente. Esse estudo analisou 454 publicações sobre o uso APF, e verificou que apenas 18 entre os trabalhos analisados sugeriram especificamente melhorias de precisão para o uso dessa técnica. Após a conclusão do estudo, um dos pontos definidos que mais apresentou problemas foi a defasagem da técnica no que se refere à evolução tecnológica dos sistemas computacionais.

A Análise de Pontos de Função tende a gerar estimativas mais precisas quando aplicada a escopos de projetos mais detalhados, na fase de análise de requisitos. ([SANTIN, 2014](#)). Além disso, a autora observou em seus estudos a importância da atualização constante dos históricos de medições, otimizando o processo de estimação.

### 3.3 Planning Poker

De acordo com [Haugen \(2006\)](#), Planning Poker é uma técnica que, quando utilizada, tende a otimizar o processo de estimaco. Em uma anlise de 101 estimaces reais, realizadas pelo mesmo time gil (equipes que aplicam metodologias geis em seus projetos), em quatro projetos diferentes, chegou-se a concluso que a utilizao do Planning Poker melhorou a performance das estimativas em comparao com tcnicas no-estruturadas de estimaco. ([HAUGEN, 2006](#)).

[Molokken-Ostvold e Haugen \(2007\)](#), por meio dos seus estudos, concluíram que o tipo de mtodo estruturado utilizado no faz diferena para a preciso do resultado das estimativas, mas sim a motivao da equipe e se a mesma definiu bem seus objetivos, característica fundamental para o sucesso da aplicao do Planning Poker.

A sequncia de Fibonacci foi utilizada por [Tamrakar e Jrgensen \(2012\)](#) como parmetro para as estimativas com o Planning Poker. Seu estudo concluiu que a escolha da escala influencia no resultado final da estimativa. Os resultados mostraram que o uso da sequncia Fibonacci tende a reduzir o valor das estimativas. Isto por que as pessoas tendem escolher valores que esto no meio da escala, quando esto em um ambiente de incerteza. Por outro lado, o valor do meio da sequncia Fibonacci tende a ser menor que em escalas lineares.

## 4 METODOLOGIA

A metodologia deste trabalho tem como enfoque a definição dos meios utilizados para a obtenção de estimativas de desenvolvimento de software, aplicando-se as métricas de software Planning Poker e Análise de Pontos de Função no contexto do desenvolvimento do software SIP. Estimou-se o esforço e a prioridade no processo de desenvolvimento do sistema.

### 4.1 Processo de Desenvolvimento com SCRUM

Por ser um projeto pequeno, o SIP foi desenvolvido por uma única pessoa, dessa forma, a mesma foi designada como *Scrum Master*. Enquanto que o papel de *Product Owner* foi destinado para a Pró-Reitora de Graduação e a Diretora de Ensino da UFVJM.

A primeira etapa do processo de desenvolvimento foi o levantamento dos requisitos, a fim de se obter as necessidades dos usuários. Nesta etapa, os *Product Owners*, com o auxílio do *Scrum Master*, redigiram as histórias de usuário, que explicam para **quem**, o **que** e **por que** estão sendo criadas as funcionalidade. Após, a análise de viabilidade das histórias de usuário, feita pelo *Scrum Master* e os *Product Owners*, foram realizadas as estimativas através das métricas de software.

As estimativas foram feitas tendo como base as histórias de usuário, que estão listadas abaixo. O termo diretoria utilizado nas histórias refere-se aos departamentos dentro da PROGRAD.

- 1 : Como pró-reitora de graduação, eu quero cadastrar diretorias como usuários, inserindo nome, senha e permissão de acesso.
- 2 : Como pró-reitora de graduação, eu quero visualizar os usuários cadastrados em uma lista e editá-los ou deletá-los.
- 3 : Como pró-reitora de graduação, eu quero buscar por usuários previamente cadastrados pelo nome.
- 4 : Como pró-reitora de graduação e diretoria, eu quero cadastrar protocolos, inserindo o remetente, o tipo do documento, o destinatário, a descrição do teor, a data de envio e a data de recebimento, além de anexar o documento em questão.
- 5 : Como pró-reitora de graduação, eu quero deletar protocolos quando necessário.
- 6 : Como pró-reitora de graduação e diretoria, eu quero visualizar os protocolos cadastrados em uma lista e editá-los.
- 7 : Como pró-reitora de graduação e diretoria, eu quero buscar por protocolos previamente cadastrados pelo remetente, destinatário ou descrição do teor.

- 8 : Como pró-reitora de graduação, eu quero cadastrar tipos de documentos, inserindo nome.
- 9 : Como pró-reitora de graduação, eu quero visualizar os tipos de documentos cadastrados em uma lista e editá-los ou deletá-los.
- 10 : Como pró-reitora de graduação, eu quero buscar por tipos de documentos previamente cadastrados pelo nome.
- 11 : Como pró-reitora de graduação e diretoria, eu quero cadastrar reuniões, inserindo os participantes, a data, a hora, o local e a pauta, além de anexar documentos se necessário.
- 12 : Como pró-reitora de graduação e diretoria, eu quero visualizar reuniões que eu fui convocada em uma lista e editar e deletar as que eu convoquei.
- 13 : Como pró-reitora de graduação e diretoria, eu quero buscar por reuniões previamente cadastradas pela data, pauta ou participantes.
- 14 : Como pró-reitora de graduação e diretoria, eu quero ser notificada caso eu tenha sido convocada para alguma reunião.
- 15 : Como pró-reitora de graduação e diretoria, eu quero ser notificada caso eu tenha recebido algum protocolo.
- 16 : Como pró-reitora de graduação e diretoria, eu quero fazer login no sistema com o nome do usuário e senha.

## 4.2 Estimativa de Software

### 4.2.1 Planning Poker

#### 4.2.1.1 Estimativa de Esforço

Para a estimativa de esforço, foram realizadas duas reuniões. Devido a uma subestimação observada nos resultados obtidos na primeira reunião, decidiu-se que uma segunda era necessária para que novas estimativas fossem geradas.

A estimativa de esforço, de acordo com [Haugen \(2006\)](#), é realizada pela equipe de desenvolvimento, juntamente com os *Product Owners*, estimando o tempo (em horas) necessário para a realização de determinada tarefa.

A estimativa de esforço para este trabalho é baseada nos números da sequência Fibonacci, representados nas cartas entregues aos membros da equipe na realização da reunião. Ao se apresentar uma história de usuário, os membros da reunião sugerem um número de horas que acreditam ser o esforço ideal a ser empregado para o desenvolvimento da referida história. Caso os números sugeridos não apresentem valores muito divergentes entre si, os mesmos são somados e divididos pelo número de membros da reunião, gerando assim uma média. Entretanto, caso haja algum valor muito discrepante dos outros, discute-se o porque do valor sugerido e

apresenta-se uma nova estimativa em relação ao valor divergente. Ao final, as médias levantadas para cada história de usuário são somadas e assim, obtêm-se o esforço total estimado para o desenvolvimento do sistema.

De acordo com [Haugen \(2006\)](#), a estimativa de esforço é melhor elucidada pelos desenvolvedores, assim como a de prioridade pelos clientes. A sequência numérica escolhida para compor as cartas foi a Fibonacci. Segundo [Tamrakar e Jørgensen \(2012\)](#), a aplicação da sequência Fibonacci pode acelerar o processo de estimação e refletir na precisão dos esforços de estimação melhor que em metodologias lineares. A sequência escolhida foi 0, 0.5, 1, 2, 3, 5, 8, 13, 21 e ?. Cada valor refere-se a horas gastas por história de usuário. O valor de 0.5 não está presente na sequência de Fibonacci original, entretanto, o mesmo foi inserido na mesma com o intuito de possibilitar a estimação de histórias de usuários simples que poderiam demandar menos de uma hora para ser desenvolvida. O sinal de interrogação integra a sequência de maneira a possibilitar que participantes da reunião possam relatar suas dúvidas ou incertezas para o resto do grupo.

#### 4.2.1.2 Estimativa de Prioridade

A estimativa de prioridade, de acordo com [Synergia \(2015\)](#), é a atribuição de valores para a ordem em que as histórias de usuário do sistema devem ser desenvolvidas, sendo avaliado a importância de cada uma. Para a estimativa de prioridade no desenvolvimento do SIP, foram escolhidas três cartas, 1 para baixa prioridade, 3 para prioridade média e 5 para alta prioridade. Para cada história de usuário apresentada, os membros da equipe atribuem um valor que define a prioridade. Caso algum valor sugerido seja diferente dos demais, a equipe discute e tenta chegar a um consenso, dessa forma, o valor final deve ser da concordância de todos. Para a estimativa de prioridade, foi necessária somente uma reunião entre a equipe de desenvolvimento e os *Product Owners*, pois durante as reuniões, os membros chegaram a um consenso em relação as estimativas.

#### 4.2.2 Análise de Pontos de Função

A Análise de Pontos de Função (APF) pode ser aplicada tanto para sistemas prontos, quanto para aqueles que ainda estão na fase de desenvolvimento. ([VAZQUEZ; SIMOES; ALBERT, 2010](#)). Sendo assim, a APF foi escolhida como métrica para a avaliação do SIP por possuir um aspecto mais técnico, em contraste com o PP, que possui um grau maior de subjetividade. Ademais, a APF pode ser aplicada de forma individual, ao contrário do PP, que deve ser aplicado em equipes, gerando assim um contraste entre as duas aplicações, enriquecendo o estudo relatado neste trabalho.

Após o levantamento dos requisitos e escrita das histórias de usuários, a aplicação da APF teve início com a compilação da documentação disponível, sendo importante que a análise da documentação dos requisitos estivesse consistente, para que a aplicação desta não

fosse prejudicada. O próximo passo foi definir o tipo de contagem. De acordo com (VAZQUEZ; SIMOES; ALBERT, 2010), existem três tipos de contagens:

- Contagem de um projeto de desenvolvimento;
- Contagem de um projeto de melhoria;
- Contagem de uma aplicação.

O tipo de contagem adotada neste trabalho foi a de um projeto de desenvolvimento, uma vez que o SIP ainda estava sendo desenvolvido quando a métrica foi aplicada. O passo seguinte foi definir o Escopo da Contagem, ou seja, o quanto do software seria avaliado. Como o software em questão era um sistema simples, optou-se pela avaliação de todas as suas funcionalidades.

O uso da métrica APF exige a determinação da Fronteira da Aplicação, que é o limite entre o sistema e o mundo real (usuário). (VAZQUEZ; SIMOES; ALBERT, 2010). Entretanto, como o SIP é um sistema independente, não é necessária a determinação da Fronteira de Aplicação.

Posteriormente, mediu-se as Funções do Tipo Dado e as Funções do Tipo Transação. As Funções do Tipo Dado são funcionalidades que atendem o usuário para o armazenamento de dados. (VAZQUEZ; SIMOES; ALBERT, 2010). No caso do SIP, o mesmo somente possui Funções do Tipo Dado classificadas como Arquivo Lógico Interno (ALI). “Um ALI representa dados mantidos pela aplicação”. (VAZQUEZ; SIMOES; ALBERT, 2010, p. 71). Um ALI pode ser exemplificado como tabelas de banco de dados atualizadas pela aplicação, dentro da fronteira interna (caso esteja especificada) da aplicação. (VAZQUEZ; SIMOES; ALBERT, 2010). Arquivo de Interface Externa (AIE) é um tipo de dado que é referenciado por uma aplicação interna e mantido por outra externa. (VAZQUEZ; SIMOES; ALBERT, 2010). Logo, este não possuía aplicação no SIP. Cada Arquivo Lógico Interno deve ser classificado de acordo com seu nível de complexidade funcional (baixa, média ou alta) e de acordo com dois parâmetros (VAZQUEZ; SIMOES; ALBERT, 2010):

- Número de Tipos de Dados (TD): pode ser exemplificado como um atributo de uma entidade em um banco de dados;
- Número de Tipos de Registros (TR): pode ser exemplificado como subgrupo de dados, ou entidade de um banco de dados.

As Funções do Tipo Transação, de acordo com Lopes e Braga (2011), são funcionalidades que compõem a base do sistema. Funções do Tipo Transação também podem ser chamadas de processos elementares, sendo estes a menor unidade funcional de um programa. Existem três tipos de Funções do Tipo Transação (VAZQUEZ; SIMOES; ALBERT, 2010):

- Entrada Externa (EE): manter (incluir, alterar e excluir) dados;

- Saída Externa (SE): objetivo de enviar arquivos para fora da fronteira da aplicação;
- Consulta Externa (CE): apresentar ao usuário uma recuperação de dados.

Primeiramente, cada processo elementar foi classificado de acordo com os tipos de Função do Tipo Transação. Ao identificar todos os processos elementares, foi necessário determinar sua complexidade, que possui dois parâmetros de medida ([VAZQUEZ; SIMOES; ALBERT, 2010](#)):

- Número de Arquivos Referenciados (AR): ALI mantido ou lido por uma Função do Tipo Transação;
- Número de Tipos de Dados.

Com o levantamento dos dados anteriores, foi possível aplicar os valores às tabelas de complexidade e de contribuição. Todas elas são padronizadas pelo IFPUG.

“O IFPUG é uma entidade sem fins lucrativos, composta por pessoas e empresas de diversos países, cuja finalidade é promover um melhor gerenciamento dos processos de desenvolvimento e manutenção de software com o uso de pontos de função e outras técnicas de medição”. ([VAZQUEZ; SIMOES; ALBERT, 2010](#), p. 36).

Tabela 1 – Tabela de complexidade funcional dos ALI e AIE.

<b>Tipos de Registros/Tipos de Dados</b>	<b>&lt;20</b>	<b>20 - 50</b>	<b>&gt;50</b>
<b>1</b>	Baixa	Baixa	Média
<b>2 - 5</b>	Baixa	Média	Alta
<b>&gt;5</b>	Média	Alta	Alta

Fonte: [Vazquez, Simoes e Albert \(2010, p. 75\)](#)

Tabela 2 – Tabela de complexidade para entradas externas (EEs).

<b>Arquivos Referenciados/Tipos de Dados</b>	<b>&lt;5</b>	<b>5 - 15</b>	<b>&gt;15</b>
<b>&lt;2</b>	Baixa	Baixa	Média
<b>2</b>	Baixa	Média	Alta
<b>&gt;2</b>	Média	Alta	Alta

Fonte: [Vazquez, Simoes e Albert \(2010, p. 112\)](#)

Tabela 3 – Tabela de complexidade para saídas externas (SEs) e consultas externas (CEs).

<b>Arquivos Referenciados/Tipos de Dados</b>	<b>&lt;6</b>	<b>6 - 19</b>	<b>&gt;19</b>
<b>&lt;2</b>	Baixa	Baixa	Média
<b>2 - 3</b>	Baixa	Média	Alta
<b>&gt;3</b>	Média	Alta	Alta

Fonte: [Vazquez, Simoes e Albert \(2010, p. 112\)](#)

Tabela 4 – Tabela de contribuição dos pontos de função não ajustados das funções do tipo dado.

<b>Tipo de Função</b>	<b>Baixa</b>	<b>Média</b>	<b>Alta</b>
Arquivo Lógico Interno	7 PF	10 PF	15 PF
Arquivo de Interface Externa	5 PF	7 PF	10 PF

Fonte: [Vazquez, Simoes e Albert \(2010, p. 78\)](#)



Tabela 5 – Tabela de contribuição dos pontos de função das funções do tipo transação.

<b>Tipo de Função</b>	<b>Baixa</b>	<b>Média</b>	<b>Alta</b>
Entrada Externa	3 PF	4 PF	6 PF
Saída Externa	4 PF	5 PF	7 PF
Consulta Externa	3 PF	4 PF	6 PF

Fonte: (VAZQUEZ; SIMOES; ALBERT, 2010, p. 116)

Ao se multiplicar o número de tipos de Função Transação pelo valor de sua referida contribuição, foi possível obter o número de pontos de função não ajustados para todo o sistema.

Segundo Vazquez, Simoes e Albert (2010), Fator de Ajuste reflete o impacto dos requisitos não funcionais no valor dos pontos de função, podendo ocasionar uma variação de até +-35%. Ainda de acordo com Vazquez, Simoes e Albert (2010), o valor do fator de ajuste é calculado de acordo com quatorze características gerais de sistema (CGS), sendo que essas são classificadas da seguinte maneira:

- 0 para nenhuma influência;
- 1 para influência mínima;
- 2 para influência moderada;
- 3 para influência média;
- 4 para influência significativa;
- 5 para grande influência.

Ainda segundo Vazquez, Simoes e Albert (2010), as 14 características gerais de sistema são:

- Comunicação de Dados: tem como objetivo descrever o nível de comunicação da aplicação com o processador;
- Processamento Distribuído: descrição do nível de distribuição dos dados entre os componentes da aplicação;
- Performance: qual o nível de influência do tempo de resposta e taxa de transações no funcionamento da aplicação;
- Configuração Altamente Utilizada: o quanto as restrições de recursos computacionais influenciam no desenvolvimento da aplicação;
- Volume de Transações: a que nível o volume de transações afeta a aplicação;
- Entrada de Dados *On-Line*: descreve o nível da entrada de dados na aplicação por meio de transações interativas;

- Eficiência do Usuário Final: o nível de como a influência sobre fatores humanos e facilidade de uso influenciam na aplicação;
- Atualização *On-Line*: o nível de atualização on-line dos arquivos lógicos internos da aplicação;
- Complexidade de Processamento: como o processamento lógico ou matemático influencia na aplicação;
- Reusabilidade: a que nível os componentes da aplicação foram projetados para serem reutilizados em outras aplicações;
- Facilidade de Instalação: descreve como o nível de dificuldade para instalação influencia na aplicação;
- Facilidade de Operação: a que nível a aplicação atende a alguns aspectos como inicialização, segurança e recuperação;
- Múltiplos Locais: o nível de heterogeneidade da aplicação, ou seja, o quanto ela está preparada para atender diversos ambientes, tanto de software quanto de hardware;
- Facilidade de Mudanças: descreve o quanto a aplicação está preparada para receber mudanças em sua lógica de processamento ou estrutura de dados.

O valor do fator de ajuste foi calculado pela fórmula:

$$VAF = (TDI \times 0,01) + 0,65$$

TDI representa o somatório dos níveis de influência das características gerais de sistema (CGS). (VAZQUEZ; SIMOES; ALBERT, 2010).

O último passo da APF é o Cálculo do Tamanho Funcional. Para sua realização, em um projeto em desenvolvimento, existe uma fórmula específica, que segundo Vazquez, Simoes e Albert (2010), é representada por:

$$DFP = (ADD + CFP)$$

- DFP: tamanho do projeto de desenvolvimento;
- ADD: tamanho das funções entregues;
- CFP: tamanho das funções de conversão.

Ao aplicar a fórmula acima, obteve-se o número de pontos de função estimados para o SIP. Com o objetivo de se estimar o esforço (em horas), empregado ao desenvolvimento de pontos de função, utilizou-se para conversão a Tabela de Produtividade Mínima da ANCINE (2008), que determina a produtividade mínima (em horas) referente a cada linguagem de programação.

Tabela 6 – Tabela de Produtividade Mínima.

<b>Desenvolvimento e manutenção de sistemas</b>	
<b>Tecnologia</b>	<b>Produtividade Mínima</b>
Java	15h/PF
ASP (Vbscript e Javascript)	10h/PF
PHP	11h/PF
JSP	13h/PF
HTML	7h/PF
ColdFusion	11h/PF
Delphi	9h/PF
Crystal reports	9h/PF
PL/SQL	9h/PF
Visual Basic	9h/PF

Fonte: ([ANCINE, 2008](#))

### 4.3 Intervalo de Confiança

Realizar uma estimativa sobre o esforço a ser empregado para o desenvolvimento de um sistema não é uma tarefa fácil. De acordo com [Ventura \(2015\)](#), existem muitas variáveis que podem influenciar na quantidade de esforço empregado para o desenvolvimento de um sistema, como arquitetura do software, linguagem de programação utilizada, modelo de ciclo de vida empregado, perfil da equipe envolvida no projeto, dentre outras ações. Ainda de acordo com [Ventura \(2015\)](#), para que possa se realizar uma boa estimativa de esforço, deve-se identificar projetos anteriores semelhantes e o esforço empregado, atualizar a contagem de pontos de função de cada projeto, etc. Caso não exista um histórico de projetos, cada variável deve ser analisada cuidadosamente e deve-se realizar um controle de risco. Uma das maneiras de se obter estimativas confiáveis em relação ao esforço que deve ser empregado no desenvolvimento de um sistema é o intervalo de confiança.

“Um intervalo de confiança é um intervalo estimado de um parâmetro de interesse de uma população. Em vez de estimar o parâmetro por um único valor, é dado um intervalo de estimativas prováveis”. ([ACTION, 2018](#)). No caso da APF, a amostra seria alguns dados coletados de horas gastas no desenvolvimento de certas funcionalidades do sistema e a população seria o tempo gasto em todas essas funcionalidades.

Amostra é um subconjunto da população e população é um conjunto de elementos que possuem alguma característica em comum. ([CORREA, 2003](#)). Neste trabalho, a amostra seria um subconjunto de tarefas e seus respectivos esforços (em horas) empregados para desenvolver certas funcionalidades do SIP. Neste caso, não é uma estimativa e sim dados coletados durante o desenvolvimento de funcionalidades do SIP. Já a população é o conjunto dos esforços empregados para desenvolver todas as funcionalidades do SIP, não sendo a mesma conhecida neste contexto. A média amostral é o valor resultante da soma de todos os elementos da amostra divididos pela quantidade dos mesmos. ([CORREA, 2003](#)). A variância “é a medida que dá o grau de dispersão

(ou de concentração) de probabilidade em torno da média” (CORREA, 2003, p. 80), sendo que, no contexto do SIP, seria em torno da média amostral. O nível de confiança caracteriza-se pela porcentagem em que a média pode estar presente no intervalo de confiança determinado. (CORREA, 2003).

Para validação das premissas, foi realizado um teste Shapiro-Wilk, presente no software R Studio. O teste em questão avalia se há evidências de normalidade em certa amostra. (REGO; BATISTA; RIBEIRO, 2017).

Para o cálculo do intervalo de confiança, é necessário que se conheça a média amostral, a variância amostral, o número de elementos presentes na amostra e o nível de confiança do intervalo. Os valores da média e variância amostrais foram calculados com o auxílio de uma calculadora científica. A fórmula utilizada para o cálculo do intervalo de confiança está representada abaixo (ACTION, 2018):

$$IC(\mu, 1 - \alpha) = \left( \bar{X} - t_{\alpha/2} \frac{s}{\sqrt{n}}; \bar{X} + t_{\alpha/2} \frac{s}{\sqrt{n}} \right)$$

$1 - \alpha$  : nível de confiança

$\bar{X}$  : média amostral

$t$  : distribuição t

$s$  : variância amostral

$n$  : número de elementos da amostra

“A distribuição t de Student é uma das distribuições mais utilizadas na estatística, com aplicações que vão desde a modelagem estatística até testes de hipóteses”. (ACTION, 2018). O valor do nível de confiança foi estabelecido em 95%, pois segundo Patino e Ferreira (2015), este valor é um dos mais comuns relatados na literatura. O valor da distribuição t de Student é identificado pelo nível de confiança, sendo que o último é o parâmetro para a identificação na tabela da Distribuição T Student, representada pela figura abaixo:

Figura 1 – Distribuição T Student.

Distribuição <i>t</i> -Student : Valores $t_c$ tais que $P(-t_c \leq t \leq t_c) = 1 - p$																
	p->90%	80%	70%	60%	50%	40%	30%	20%	10%	5%	4%	2%	1%	0.2%	0.1%	
1	0,158	0,325	0,510	0,727	1,000	1,376	1,963	3,078	6,314	12,706	15,894	31,821	63,656	318,289	636,578	1
2	0,142	0,289	0,445	0,617	0,816	1,061	1,386	1,886	2,920	4,303	4,849	6,965	9,925	22,328	31,600	2
3	0,137	0,277	0,424	0,584	0,765	0,978	1,250	1,638	2,353	3,182	3,482	4,541	5,841	10,214	12,924	3
4	0,134	0,271	0,414	0,569	0,741	0,941	1,190	1,533	2,132	2,776	2,999	3,747	4,604	7,173	8,610	4
5	0,132	0,267	0,408	0,559	0,727	0,920	1,156	1,476	2,015	2,571	2,757	3,365	4,032	5,894	6,869	5
6	0,131	0,265	0,404	0,553	0,718	0,906	1,134	1,440	1,943	2,447	2,612	3,143	3,707	5,208	5,959	6
7	0,130	0,263	0,402	0,549	0,711	0,896	1,119	1,415	1,895	2,365	2,517	2,998	3,499	4,785	5,408	7
8	0,130	0,262	0,399	0,546	0,706	0,889	1,108	1,397	1,860	2,306	2,449	2,896	3,355	4,501	5,041	8
9	0,129	0,261	0,398	0,543	0,703	0,883	1,100	1,383	1,833	2,262	2,398	2,821	3,250	4,297	4,781	9
10	0,129	0,260	0,397	0,542	0,700	0,879	1,093	1,372	1,812	2,228	2,359	2,764	3,169	4,144	4,587	10
11	0,129	0,260	0,396	0,540	0,697	0,876	1,088	1,363	1,796	2,201	2,328	2,718	3,106	4,025	4,437	11
12	0,128	0,259	0,395	0,539	0,695	0,873	1,083	1,356	1,782	2,179	2,303	2,681	3,055	3,930	4,318	12
13	0,128	0,259	0,394	0,538	0,694	0,870	1,079	1,350	1,771	2,160	2,282	2,650	3,012	3,852	4,221	13
14	0,128	0,258	0,393	0,537	0,692	0,868	1,076	1,345	1,761	2,145	2,264	2,624	2,977	3,787	4,140	14
15	0,128	0,258	0,393	0,536	0,691	0,866	1,074	1,341	1,753	2,131	2,249	2,602	2,947	3,733	4,073	15
16	0,128	0,258	0,392	0,535	0,690	0,865	1,071	1,337	1,746	2,120	2,235	2,583	2,921	3,686	4,015	16
17	0,128	0,257	0,392	0,534	0,689	0,863	1,069	1,333	1,740	2,110	2,224	2,567	2,898	3,646	3,965	17
18	0,127	0,257	0,392	0,534	0,688	0,862	1,067	1,330	1,734	2,101	2,214	2,552	2,878	3,610	3,922	18
19	0,127	0,257	0,391	0,533	0,688	0,861	1,066	1,328	1,729	2,093	2,205	2,539	2,861	3,579	3,883	19
20	0,127	0,257	0,391	0,533	0,687	0,860	1,064	1,325	1,725	2,086	2,197	2,528	2,845	3,552	3,850	20
21	0,127	0,257	0,391	0,532	0,686	0,859	1,063	1,323	1,721	2,080	2,189	2,518	2,831	3,527	3,819	21
22	0,127	0,256	0,390	0,532	0,686	0,858	1,061	1,321	1,717	2,074	2,183	2,508	2,819	3,505	3,792	22
23	0,127	0,256	0,390	0,532	0,685	0,858	1,060	1,319	1,714	2,069	2,177	2,500	2,807	3,485	3,768	23
24	0,127	0,256	0,390	0,531	0,685	0,857	1,059	1,318	1,711	2,064	2,172	2,492	2,797	3,467	3,745	24
25	0,127	0,256	0,390	0,531	0,684	0,856	1,058	1,316	1,708	2,060	2,167	2,485	2,787	3,450	3,725	25
26	0,127	0,256	0,390	0,531	0,684	0,856	1,058	1,315	1,706	2,056	2,162	2,479	2,779	3,435	3,707	26
27	0,127	0,256	0,389	0,531	0,684	0,855	1,057	1,314	1,703	2,052	2,158	2,473	2,771	3,421	3,689	27
28	0,127	0,256	0,389	0,530	0,683	0,855	1,056	1,313	1,701	2,048	2,154	2,467	2,763	3,408	3,674	28
29	0,127	0,256	0,389	0,530	0,683	0,854	1,055	1,311	1,699	2,045	2,150	2,462	2,756	3,396	3,660	29
30	0,127	0,256	0,389	0,530	0,683	0,854	1,055	1,310	1,697	2,042	2,147	2,457	2,750	3,385	3,646	30
35	0,127	0,255	0,388	0,529	0,682	0,852	1,052	1,306	1,690	2,030	2,133	2,438	2,724	3,340	3,591	35
40	0,126	0,255	0,388	0,529	0,681	0,851	1,050	1,303	1,684	2,021	2,123	2,423	2,704	3,307	3,551	40
50	0,126	0,255	0,388	0,528	0,679	0,849	1,047	1,299	1,676	2,009	2,109	2,403	2,678	3,261	3,496	50
60	0,126	0,254	0,387	0,527	0,679	0,848	1,045	1,296	1,671	2,000	2,099	2,390	2,660	3,232	3,460	60
120	0,126	0,254	0,386	0,526	0,677	0,845	1,041	1,289	1,658	1,980	2,076	2,358	2,617	3,160	3,373	120
∞	0,126	0,253	0,385	0,524	0,675	0,842	1,036	1,282	1,645	1,960	2,054	2,327	2,576	3,091	3,291	∞

Fonte: UNB (2018)



## 5 RESULTADOS

Os resultados da aplicação das métricas PP e APF no desenvolvimento do SIP foram obtidos considerando-se que o desenvolvedor trabalhava cerca de quatro horas por dia, cinco dias por semana.

### 5.1 Análise de Pontos de Função

A aplicação da APF ocorreu durante cinco dias, sendo que informações previamente coletadas, referente à métrica, foram melhor estudadas, juntamente com a análise das histórias de usuário e determinação dos processos elementares. A partir da definição dos processos elementares, foi possível aplicar os parâmetros definidos pela APF. As tabelas a seguir explicitam as informações coletadas de acordo com a análise das Funções do Tipo Dado e Funções do Tipo Transação.

Tabela 7 – Resultados da aplicação da APF.

<b>Descrição da Função</b>	<b>Tipo</b>	<b>TD</b>	<b>AR/TR</b>	<b>Complexidade</b>	<b>Contribuição (em pontos de função)</b>
Usuário	ALI	4	1	Baixa	7 PF
Usuário - cadastro	EE	5	1	Baixa	3 PF
Usuário - alteração	EE	9	1	Baixa	3 PF
Usuário - edição (consulta banco de dados)	CE	4	1	Baixa	3 PF
Usuário - exclusão	EE	3	1	Baixa	3 PF
Usuário - consulta	CE	2	1	Baixa	3 PF
Protocolo	ALI	7	2	Baixa	7 PF
Protocolo - cadastro	EE	9	3	Alta	6 PF
Protocolo - alteração	EE	11	3	Alta	6 PF
Protocolo - edição (consulta banco de dados)	CE	7	3	Média	4 PF
Protocolo - exclusão	EE	3	2	Baixa	3 PF
Protocolo - consulta	CE	4	2	Baixa	3 PF

Tabela 8 – Resultados da aplicação da APF.

<b>Descrição da Função</b>	<b>Tipo</b>	<b>TD</b>	<b>AR/TR</b>	<b>Complexidade</b>	<b>Contribuição (em pontos de função)</b>
Tipo de documento	ALI	1	1	Baixa	7 PF
Tipo de documento - cadastro	EE	2	1	Baixa	3 PF
Tipo de documento - alteração	EE	6	1	Baixa	3 PF
Tipo de documento - edição (consulta banco de dados)	CE	1	1	Baixa	3 PF
Tipo de documento - exclusão	EE	3	1	Baixa	3 PF
Tipo de documento - consulta	CE	2	1	Baixa	3 PF
Reunião	ALI	6	2	Baixa	7 PF
Reunião - agendamento	EE	8	2	Média	4 PF
Reunião - alteração	EE	11	2	Média	4 PF
Reunião - edição (consulta banco de dados)	CE	6	2	Média	4 PF
Reunião - exclusão	EE	3	2	Baixa	3 PF
Reunião - consulta	CE	4	2	Baixa	3 PF
Reunião - notificação	CE	5	2	Baixa	3 PF
Protocolo - notificação	CE	4	1	Baixa	3 PF
Login	CE	3	1	Baixa	3 PF



Tabela 9 – Tabela de Contribuição das Funções do Tipo Dado e do Tipo Transação.

<b>Tipo de Função</b>	<b>Complexidade Funcional</b>	<b>Totais por Tipo de Complexidade</b>	<b>Totais por Tipo de Função</b>
ALI	4 Baixa x7 0 Média x10 0 Alta x15	= 28 = 0 = 0	28
EE	8 Baixa x3 2 Média x4 2 Alta x6	= 24 = 8 = 12	38
CE	9 Baixa x3 2 Média x4 0 Alta x6	= 28 = 8 = 0	36
<b>Total</b>			102

Para o cálculo do número de Tipos de Dado relacionados aos Arquivos Lógicos Internos (ALI), foram contabilizados campos de inserção de dados (input). Sendo assim, *Usuário* foi definido como ALI pelo fato de representar um grupo de dados, assim como *Protocolo*, *Tipo de Documento* e *Reunião*. No cálculo do número de Tipos de Dado (TD) relacionados a Funções do Tipo Transação, foram contabilizados campos de preenchimento de dados e, quando pertinente, os botões envolvidos no processo. *Usuário - cadastro* foi definido como EE (Entrada Externa) pelo fato de representar a inserção de dados novos na aplicação. *Usuário - consulta* foi definido como CE (Consulta Externa) pelo fato de o sistema realizar uma consulta ao banco de dados para retornar os valores desejados, que são exibidos em uma lista, na tela do sistema, contendo todos os usuários cadastrados. *Usuário - alteração* refere-se à função de edição de dados do usuário previamente cadastrado, incluindo na contagem de TD os campos e botões envolvidos na operação. *Usuário - edição (consulta ao banco de dados)* refere-se à busca no banco de dados que o sistema realiza para que os dados do usuário previamente cadastrado possam ser alterados (tela de edição). *Usuário - exclusão* refere-se à função de deletar o usuário previamente cadastrado. *Usuário - consulta* refere-se a lista de visualização de todos os usuários previamente cadastrados. *Protocolo*, *Tipo de Documento* e *Reunião* receberam basicamente os mesmos processos elementares. Entretanto, o ALI *Reunião* recebeu dois processos elementares a mais, sendo eles *Reunião - notificação*, uma caixa de diálogo que avisa ao usuário caso ele seja convocado para alguma reunião, e *Protocolo - notificação*, uma caixa de diálogo que avisa ao usuário caso ele receba algum protocolo, contabilizando os TDs mostrados e os botões disponíveis. O processo elementar *Login* foi definido como CE pelo fato de ser uma consulta ao banco de dados.

Os TRs foram contabilizados de acordo com as regras de contagem de [Vazquez, Simoes e Albert \(2010\)](#), que determina que “cada função de dados tem um subgrupo de TD a ser contado como um TR”. ([VAZQUEZ; SIMOES; ALBERT, 2010](#), p. 78). Os ARs também foram contabilizados de acordo com as regras de contagem de [Vazquez, Simoes e Albert \(2010\)](#), que determina que a cada ALI deve ser contabilizado um AR.

Os processos elementares tiveram suas complexidades definidas por meio das tabelas de contribuição. A análise das Funções do Tipo Dado e do Tipo Transação resultaram em um número total de 102 pontos de função não ajustados. Este número foi obtido por meio da aplicação da Fórmula do Projeto de Desenvolvimento, definida por [Vazquez, Simoes e Albert \(2010\)](#):

$$DFP = ADD + CFP$$

DFP representa o tamanho do projeto de desenvolvimento, ADD indica o tamanho das funções que fazem parte do funcionamento do sistema (histórias de usuários) e CFP indica o tamanho das funções de conversão, ou seja, aquelas que só existem para configurar o sistema inicialmente. Como o projeto do SIP não exigiu funções de conversão, o cálculo do DFP foi a soma dos ADDs.

Os níveis de influência de cada característica geral do sistema foram determinados de acordo com as histórias de usuário do SIP e seguindo as Diretrizes para Determinação do Nível de Influência. ([VAZQUEZ; SIMOES; ALBERT, 2010](#)). A tabela abaixo retrata a aplicação do Fator de Ajuste nas características gerais do sistema, onde cada característica recebeu uma pontuação de 0 a 5 de acordo com a seu grau de relevância.

Tabela 10 – Tabela de Aplicação do Fator de Ajuste.

<b>Características Gerais de Sistemas</b>	<b>Nível de Influência</b>
Comunicação de Dados	5
Processamento Distribuído	0
Performance	0
Configuração Altamente Utilizada	0
Volume de Transações	0
Entrada de Dados On-line	5
Eficiência do Usuário Final	2
Atualização On-line	2
Complexidade de Processamento	0
Reusabilidade	4
Facilidade de Instalação	0
Facilidade de Operação	1
Múltiplos Locais	3
Facilidade de Mudanças	4

A Comunicação de Dados recebeu pontuação máxima pelo fato do SIP ser mais que uma aplicação front-end e oferecer mais de um protocolo como meio de comunicação. Como o SIP não é um sistema distribuído, então a característica Processamento Distribuído recebeu pontuação 0. A característica Performance recebeu pontuação 0 pelo fato do SIP ser um sistema simples e o *Product Owner* não ter requisitado desempenho. Já a Configuração Altamente Utilizada recebeu pontuação 0 pelo fato dos requisitos não especificarem restrições operacionais ao sistema. Devido a simplicidade do SIP e sua utilização por poucos usuários, o que não ocasiona picos de utilização, o Volume de Transações recebeu pontuação 0. O SIP é um sistema Web, sendo assim seu fluxo de dados *on-line* é grande, refletindo em uma pontuação 5 para Entrada de Dados On-line. Eficiência do Usuário Final recebeu pontuação 2 pelo fato do SIP possuir mecanismos que facilitem a sua utilização como:

- “Auxílio para navegação, como, por exemplo, teclas de função, saltos, menus gerados dinamicamente”; (VAZQUEZ; SIMOES; ALBERT, 2010, p. 127);
- Menus;
- Interface de mouse;
- Janelas *pop-up*.

Devido ao fato de quatro ALIs serem atualizados *on-line*, a característica Atualização *On-line* recebeu pontuação 2. Já em relação à Complexidade de Processamento, o mesmo recebeu pontuação 0 em razão do sistema não requerer processamentos complexos. Reusabilidade recebeu pontuação 4, uma vez que o SIP possui um alto grau de reutilização. Como se trata de um sistema *on-line*, que não requer instalação, a característica Facilidade de Instalação recebeu pontuação 0. Facilidade de Operação recebeu pontuação 1, levando em consideração a exigência

do salvamento, inicialização e recuperação por parte do operador do SIP. A pontuação 3 foi atribuída a Múltiplos Locais, devido a aplicação ser idealizada para operar em vários locais (hardware e software) no processo de desenvolvimento. Por último, Facilidade de Mudanças recebeu pontuação 4, pois o SIP possui dois dos parâmetros abordados pela característica geral do sistema que são, segundo [Vazquez, Simoes e Albert \(2010\)](#):

- “São fornecidos mecanismos de consulta flexível, que permitem a manipulação de pedidos complexos; por exemplo, lógica de e/ou combinadas em um ou mais arquivos lógicos (conte como três itens)”. ([VAZQUEZ; SIMOES; ALBERT, 2010](#), p. 132). Procedimentos lógicos do SIP relativos a *Protocolo* envolvem os ALIs *Usuário* e *Tipo de Documento*. Sendo assim, são três os arquivos lógicos combinados em procedimentos.
- “Dados do controle do negócio são mantidos pelo usuário por meio de processos interativos, e as alterações têm efeito imediato (conte como dois itens)”. ([VAZQUEZ; SIMOES; ALBERT, 2010](#), p. 132). Controle de negócio refere-se a manutenção de tabelas de forma *on-line*. Quando os dados são alterados ou inseridos no sistema, os mesmos são atualizados instantaneamente.

A soma dos níveis de influência resultou num total de 26 pontos. Para o cálculo do Fator de Ajuste, utilizou-se a seguinte fórmula, definida por [Vazquez, Simoes e Albert \(2010\)](#):

$$VAF = (TDI * 0,01) + 0,65$$

Onde VAF indica o valor do fator de ajuste e TDI indica o somatório dos níveis de influência das características gerais. Aplicando-se a fórmula acima, obteve-se um valor de 0,91. Para o cálculo dos pontos de função ajustados, utilizou-se a fórmula abaixo, definida por [SANTIN \(2014\)](#):

$$PFD = DFP \times VAF$$

PFD indica o número de pontos de função de desenvolvimento (ajustados). Multiplicando o VAF igual a 0,91 pelos 102 pontos de função não ajustados, obteve-se um valor de 92,82 pontos de função ajustados.

Aplicando-se o valor dos pontos de função ajustados à Tabela de Produtividade Mínima da [ANCINE \(2008\)](#) para PHP e HTML, obteve-se os valores aproximados de 13 e 8 meses, respectivamente. Como as duas linguagens de programação possuem grande importância para o desenvolvimento, foi feita uma média com as duas estimativas, obtendo-se o valor de 10 meses e meio de trabalho.

### 5.1.1 Aplicando Intervalo de Confiança à Análise de Pontos de Função

Uma dificuldade em relação à aplicação da métrica APF para estimar o tempo gasto no desenvolvimento é estabelecer quanto tempo é gasto para desenvolver cada ponto

de função. Este trabalho, inicialmente, adotou a tabela da [ANCINE \(2008\)](#) como referência. Entretanto, durante o estudo e aplicação da metodologia de Pontos de Função, foi possível notar uma diferença muito grande entre os resultados da APF e o valor real obtido. Uma possível explicação para essa diferença é que o tempo gasto no desenvolvimento de um sistema pode estar relacionado a diversos fatores como: a quantidade de artefatos gerados no processo de desenvolvimento, o tamanho da equipe de desenvolvimento, dentre outros. Assim, para uma estimativa de tempo mais precisa, utilizando APF, o mais adequado é calcular o tempo médio de gasto por ponto de função para cada equipe de desenvolvimento. Por isso, o uso de intervalo de confiança foi aplicado, neste trabalho, com o objetivo de alcançar uma melhor estimativa para a APF, que seja mais adequada à metodologia de desenvolvimento utilizada.

O Intervalo de Confiança é determinado por um intervalo de valores que estabelecem um limite, de acordo com um nível de confiança, em que a média populacional pode se encontrar. ([CORREA, 2003](#)). “Uma maneira de expressar a precisão da estimação é estabelecer limites que, com certa probabilidade, incluam o verdadeiro valor do parâmetro da população”. ([CORREA, 2003](#), p. 97).

A Tabela 11 representa a amostra coletada referente ao tempo gasto para desenvolvimento de algumas funcionalidades do SIP:

Tabela 11 – Amostra de tempo gasto para desenvolvimento de algumas funcionalidades do SIP.

<b>Descrição da Função</b>	<b>Contribuição (em pontos de função)</b>	<b>Horas gastas para cada função</b>	<b>Horas por ponto de função</b>
Usuário	7 PF	16	2,28
Usuário - cadastro	3 PF	12	4
Usuário - alteração	3 PF	11	3,67
Usuário - edição (consulta banco de dados)	3 PF	20	6,67
Usuário - exclusão	3 PF	10	3,30
Usuário - consulta	3 PF	12	4
Protocolo	7 PF	13	1,86
Protocolo - cadastro	4 PF	10	2,5
Protocolo - alteração	4 PF	9	2,25
Protocolo - edição (consulta banco de dados)	4 PF	17	4,25

Para a validação das premissas, foi levado em consideração um nível de confiança de 95%. O teste Shapiro-Wilk, realizado para validação das premissas, utiliza a estatística W, onde os elementos da amostra são ordenados de forma crescente. De acordo com [Rego, Batista e Ribeiro \(2017\)](#), o p-valor, quando maior que o nível de confiança, no caso do trabalho, de 0,05, significa a existência de evidências que a amostra provêm de uma população normal. Os resultados do teste estão representados na figura abaixo:

Figura 2 – Resultados teste de normalidade.

```
Amostra <- c(2.28, 4, 3.67, 6.67, 3.3, 4, 1.86, 2.5, 2.25, 4.25)
t.test(Amostra, conf.level = 0.95)

##
## One Sample t-test
##
## data: Amostra
## t = 7.7953, df = 9, p-value = 2.721e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  2.468697 4.487303
## sample estimates:
## mean of x
##      3.478
```

Fonte: Próprio autor.

Figura 3 – Resultados teste de normalidade.

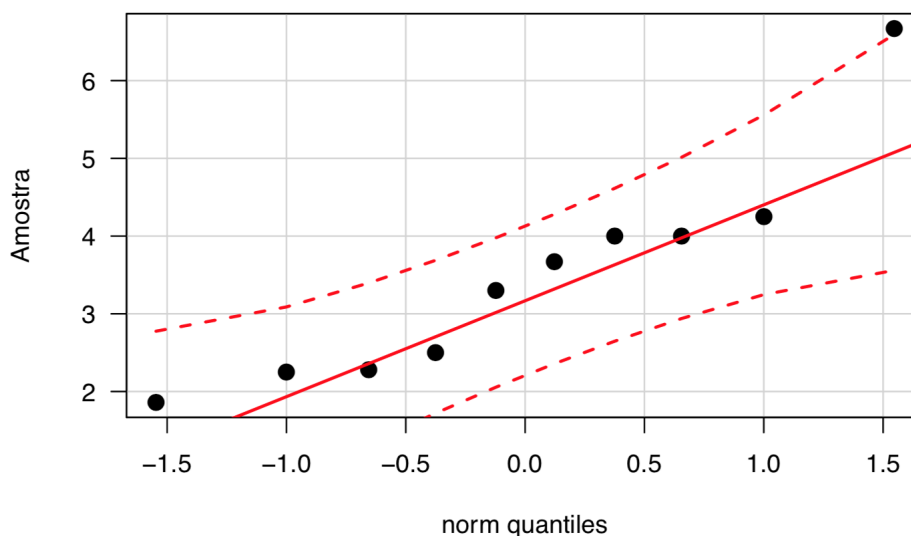
```
shapiro.test(Amostra)

##
## Shapiro-Wilk normality test
##
## data: Amostra
## W = 0.88647, p-value = 0.1547
```

Fonte: Próprio autor.

As evidências de normalidade da amostra são representadas também no gráfico, presente na figura abaixo:

Figura 4 – Gráfico de Normalidade.



Fonte: Próprio autor.

Considerando que a amostra tem 10 elementos, obteve-se 9 graus de liberdade. O mesmo é calculado subtraindo 1 do número total de elementos da amostra. (CORREA, 2003). O nível de confiança utilizado para o cálculo foi 95%, mas este pode variar de acordo com a exigência da empresa ou do projeto com relação à precisão da média populacional. De acordo com a tabela T Student, o valor de t para o grau de liberdade 9 e intervalo de confiança 0,05 é 2,262. A média amostral é 3,478 e a variância amostral é 1,410. Aplicando-se os valores encontrados na fórmula, obteve-se o seguinte intervalo de confiança:

$$IC(\mu, 0.95) = [2.469; 4.486]$$

Dessa forma, pode-se afirmar, com 95% de confiança, que o valor médio de horas por ponto de função na metodologia utilizada neste trabalho está no intervalo [2.469 ; 4.486].

Considera-se que o custo de um ponto de função pode variar de acordo com metodologia utilizada. Esse intervalo de confiança pode ser utilizado para estimar o desenvolvimento de novas funcionalidades com mais precisão.

Devido à percepção do desenvolvedor, notando que as linguagens de programação PHP, JavaScript e HTML foram empregadas no SIP de forma parecida, ou seja, a quantidade de tempo empregado para desenvolver as funcionalidades foi semelhante para todas as linguagens, assumiu-se o mesmo valor de horas por ponto de função para as três. No caso do SIP, são 92,82 pontos de função ajustados. Assim, aplicando-se os 92,82 pontos de função ajustados no intervalo de confiança obtido, obteve-se um intervalo de tempo de, aproximadamente, 229-416 horas de trabalho.

## 5.2 Planning Poker

A aplicação da técnica de Planning Poker foi realizada em duas reuniões. Devido ao fato de apenas uma pessoa estar envolvida no desenvolvimento do SIP, foi necessário convidar mais membros para participar das reuniões. Neste caso, alunos do curso de Sistemas de Informação da UFVJM, com o objetivo de agregar mais conhecimento. A primeira reunião foi composta por seis membros, sendo dois deles os *Product Owners*, neste caso a Pró-reitora de Graduação e a Diretora de Ensino da UFVJM. Os outros quatro membros eram alunos do curso de Sistemas de Informação, sendo um deles o próprio desenvolvedor do sistema. Três dos alunos do curso de Sistemas de Informação já conheciam a técnica de Planning Poker, incluindo o desenvolvedor. Os dois *Product Owners* não conheciam a técnica. As cartas utilizadas durante as reuniões continham os valores a seguir: ?, 0, 0.5, 1, 2, 3, 5, 8, 13 e 21, seguindo a lógica da sequência Fibonacci.

A primeira reunião teve duração de aproximadamente uma hora e foram levantadas estimativas de esforço, em horas, para as histórias de usuário. O valor do esforço para cada história foi calculado somando todos os valores estimados pelos participantes e calculando uma média destes valores.

Tabela 12 – Histórias de Usuário e respectivas pontuações coletadas na primeira reunião.

História de Usuário	Pontuação (em horas)
1	6,5
2	2,84
3	2,17
4	13
5	2,7
6	2,84
7	3,67
8	1,83
9	1,83
10	11,84
11	10,5
12	5,67
13	3,67
14	10,5
15	7,5
<b>Total</b>	<b>87,06</b>

A estimativa total de esforço obtida por meio da realização da primeira reunião foi de 87,06 horas, resultando em aproximadamente 21 dias de trabalho.

A segunda reunião durou cerca de uma hora e meia, sendo composta por cinco membros, quatro eram os mesmos alunos de Sistemas de Informação e um dos *Product Owner* presente na primeira reunião. Buscando maior precisão no levantamento do valores de esforço, as histórias de usuário foram divididas em tarefas. Na tabela abaixo estão os valores médios de cada



estimativa referente a cada tarefa. A segunda reunião foi dividida em duas partes. A primeira resultou em uma estimativa de 283,2 horas de trabalho, representando aproximadamente 71 dias de trabalho.

Tabela 13 – Resultados da estimativa de esforço para tarefas na segunda reunião.

<b>História de Usuário</b>	<b>Tarefas</b>	<b>Pontuação (esforço)</b>
1	Criar classe Usuário no banco de dados.	4,2
	Criar tela cadastro Usuário.	8
2	Criar tela gerenciamento Usuário.	7,4
	Criar tela editar Usuário.	5,8
	Criar consulta Usuário no banco de dados.	3,4
3	Criar função buscar Usuário.	3,2
4	Criar Classe Protocolo no banco de dados.	7,4
	Criar tela cadastro Protocolo.	13
	Criar função anexar documento Protocolo.	10
5	Criar nível de acesso para deletar Protocolo.	11
	Criar função deletar Protocolo.	4,2
6	Criar tela gerenciamento Protocolo.	9
	Criar tela editar Protocolo.	6,2
	Criar consulta Protocolo no banco de dados.	5
7	Criar função buscar Protocolo.	6,8
8	Criar classe Tipo de Documento no banco de dados.	6,2
	Criar tela cadastro Tipo de Documento.	6,8
9	Criar nível de acesso para deletar Tipo de Documento.	3,6
	Criar tela gerenciamento Tipo de Documento.	5,6
	Criar tela editar Tipo de Documento.	4,6
	Criar consulta Tipo de Documento no banco de dados.	6,2
10	Criar função buscar Tipo de Documento.	4,2
11	Criar classe Reunião no banco de dados.	7,8
	Criar tela cadastro Reunião.	11
	Criar função anexar documento Reunião.	4,8
	Criar função calendário Reunião.	21
12	Criar tela gerenciamento Reunião.	14,6
	Criar tela editar Reunião.	6,2
	Criar consulta Reunião no banco de dados.	8
13	Criar função buscar Reunião.	5
14	Criar função notificação Reunião.	21
15	Criar função notificação Protocolo.	13
16	Criar função login.	21
16	Criar tela login.	8
<b>Total</b>		<b>283,2</b>

Já na segunda parte da reunião, foi feita uma estimativa de prioridade referente a cada história de usuário, sendo o foco as necessidades do *Product Owner*. Como as tarefas são muito técnicas e seguem uma ordem pré-definida para serem desenvolvidas, foram escolhidas as histórias de usuário como parâmetros de estimativa de prioridade. Para estimar as prioridades, foram escolhidas três cartas como parâmetro, 1, 3 e 5, sendo 1 para baixa prioridade, 3 para média prioridade e 5 para alta prioridade. Em seguida, o *Product Owner*, com o auxílio dos participantes da reunião, definiu as prioridades de desenvolvimento dos quatro ALIs do sistema, sendo estes Usuário, Protocolo, Tipo de Documento e Reunião. Além disso, foi inserida como estória de usuário a funcionalidade *Login*, buscando assim mais precisão nas estimativas.

Desta forma, a ordem de prioridade definida foi: Protocolo, Tipo de Documento, Usuário e Reunião. As histórias de usuário e suas respectivas estimativas de prioridade estão retratados na tabela abaixo:

Tabela 14 – Histórias de Usuário e respectivas pontuações para prioridade coletadas na segunda reunião.

<b>História de Usuário</b>	<b>Prioridade</b>
1	5
2	1
3	3
4	5
5	1
6	1
7	3
8	5
9	1
10	3
11	5
12	1
13	3
14	1
15	3
16	1

## 6 DISCUSSÃO

Para a realização da análise dos resultados, foi levado em consideração a bibliografia utilizada no trabalho, as duas métricas de software APF e PP, o intervalo de confiança, as variações de resultados entre as aplicações das métricas, os parâmetros de medição empregados, a metodologia desenvolvida para personalizar o esforço gasto por ponto de função e o ambiente de trabalho em que estava inserido o desenvolvedor. Abaixo está representada a tabela com todos os tempos obtidos:

Tabela 15 – Resultados obtidos.

<b>Método</b>	<b>Tempo Real</b>	<b>APF</b>	<b>IC</b>	<b>PP 1ª Reunião</b>	<b>PP 2ª Reunião</b>
<b>Tempo (em horas)</b>	340	837	229-416	87	283

### 6.1 Análise de Pontos de Função

A aplicação da APF, tendo como parâmetro a tabela da [ANCINE \(2008\)](#), resultou em dez meses e meio de trabalho. Tendo em vista a carga horária exercida pelo desenvolvedor, e a complexidade do SIP, chegou-se a conclusão que o resultado encontrado estava superestimado. A principal causa desta superestimação se deve ao fato de que o desenvolvimento de sistemas de informação ser uma atividade que agrega particularidades de cada projeto, como tempo e conhecimento. Sendo assim, torna-se muito difícil padronizar ou medir a quantidade de esforço necessário para desenvolver sistemas computacionais, mesmo estes sendo similares. A superestimação se deve, também, ao fato de que alguns parâmetros definidos pelo IFPUG e, que de acordo com o próprio [Vazquez, Simoes e Albert \(2010\)](#), encontrarem-se defasados.

Para a determinação do Valor de Ajuste, foram analisadas as características gerais do sistema. Durante o estudo, foi identificada uma incoerência em uma das características gerais do sistema. Apesar da Atualização *On-line* determinar que a pontuação 5 deve ser atribuída para sistemas que possuem uma preocupação com custos de processo de recuperação e alto volume de processamento (não sendo o caso do SIP), este não se encaixa na pontuação 3 pelo fato de que todos os seus arquivos serem atualizados *on-line*. sendo que a pontuação 3 determina que a maioria dos arquivos internos possuem atualização *on-line*.

Sendo assim, a APF mostrou-se ineficiente quando aplicada por pessoas com conhecimento insuficiente a respeito da métrica, devido ao alto grau de complexidade de seus parâmetros. Nesses casos, pode ser que a estimativa demande tempo além do esperado, exigindo possíveis treinamentos aos que irão aplicá-la.

Apesar dos problemas acima relatados, a APF é uma boa solução para estimar o esforço necessário para o desenvolvimento de sistemas, de uma forma mais técnica e precisa,

quando seus parâmetros encontram-se atualizados e os aplicadores estão munidos de uma fonte de conversão (pontos de função para horas de trabalho/custo) pertinente à realidade do projeto. Durante a aplicação da métrica, foi possível perceber as especificidades dos parâmetros, assim como o detalhamento dos mesmos, que possibilitou a agregação de valor e credibilidade às estimativas. Além disso, uma grande vantagem da métrica é que ela pode ser aplicada de forma individual, não exigindo a criação de equipes.

A utilização do intervalo de confiança para estimar o esforço necessário para se desenvolver o sistema mostrou-se bem confiável e preciso, devido ao uso difundido da técnica e a similaridade dos valores encontrados em comparação com os resultados da aplicação do PP e os valores reais obtidos.

## 6.2 Planning Poker

Posterior à aplicação do PP, chegou-se a conclusão de que houve uma subestimação dos esforços necessários para o desenvolvimento do SIP, sendo que, de acordo com a complexidade do sistema e a realidade do ambiente de desenvolvimento em questão, provavelmente o mesmo demandaria mais tempo para ser concebido. A subestimação do esforço era, de certa forma, esperada pelo fato dos participantes da reunião não possuírem experiência suficiente com o desenvolvimento desse tipo de aplicação, dificultando a estimação, de forma precisa, do esforço necessário para o desenvolvimento.

Posterior à análise dos resultados da primeira reunião, foi decidido que uma segunda reunião seria necessária para obtenção de novas estimativas de esforço, de forma que as histórias de usuário foram divididas em tarefas. Ainda que para [Torrecilla-Salinas et al. \(2015\)](#), o intuito inicial da técnica de Planning Poker era substituir a estimativa de tarefas por histórias de usuário, entretanto, na prática, a estimativa de histórias mostrou-se menos precisa. Isto justifica-se pelo fato das estimativas coletadas durante primeira reunião resultarem em uma subestimação dos valores. Sendo assim, utilizando-se como parâmetro de estimação as tarefas, pôde-se observar que os valores obtidos estavam mais condizentes com o ambiente de desenvolvimento e suas particularidades.

Na segunda reunião, além da estimativa de esforço, foi realizada a estimativa de prioridade. O *Product Owner* foi o responsável por determinar a prioridade para cada história, e os membros da reunião analisavam a viabilidade da escolha e aconselhavam o mesmo na tomada de decisão. A estimativa de prioridade proporcionou que o *Product Owner* contribuísse de forma mais ativa nas decisões, sendo perceptível uma maior satisfação por parte do mesmo. Entretanto, foi inevitável a interferência do desenvolvedor e dos outros participantes nas decisões do *Product Owner*, dificultando na realização de uma análise imparcial das opiniões do último.

Apesar dos problemas relatados, a aplicação do Planning Poker mostrou-se muito eficaz para integrar os participantes e suas ideias divergentes, característica presente na metodologia ágil Scrum. Em adição, várias ideias de melhorias para o SIP foram propostas, inclusive por parte dos *Product Owners*, que não possuíam conhecimento na área de desenvolvimento de

software. Notou-se que, devido a baixa complexidade do PP, não foi preciso um extenso estudo para sua aplicação.



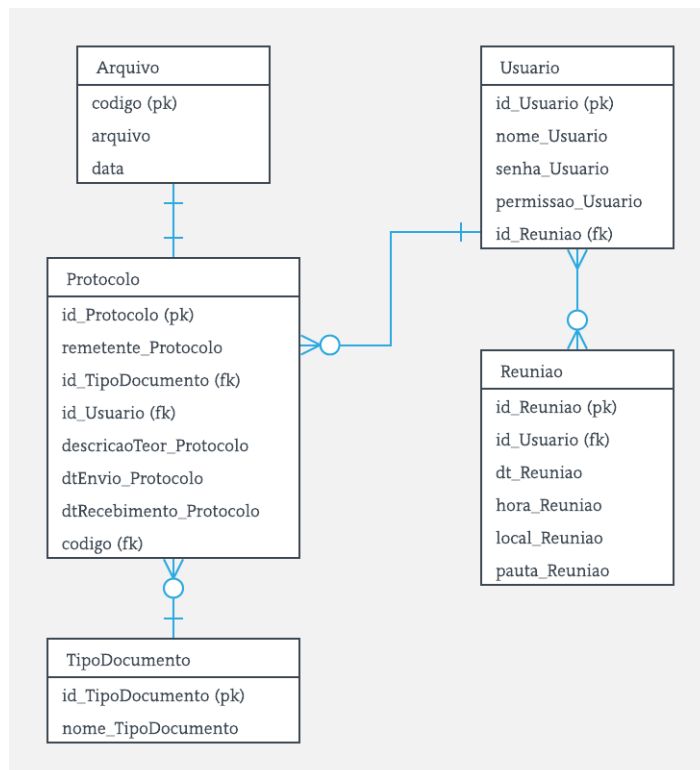
## 7 O SISTEMA

Foi proposto ao estagiário da Pró-Reitoria de Graduação (PROGRAD) que se desenvolve-se um sistema Web que iria armazenar os documentos que transitam pela mesma. O objetivo do sistema seria de possibilitar um melhor gerenciamento dos protocolos. Além disso, foi proposto que o software oferecesse a opção de agendamento de reuniões internas.

O primeiro passo para o processo de desenvolvimento do software foi a realização de reuniões com as partes interessadas, com o objetivo de coletar requisitos para o mesmo. Ficou a critério do estagiário a escolha das ferramentas que seriam utilizadas para o desenvolvimento do sistema. Assim sendo, foram escolhidas as linguagens de programação: HTML na versão 5, PHP na versão 5.6.20, JavaScript e CSS. Para complementar o processo de desenvolvimento, foi escolhido o *framework* Bootstrap para compor a interface do SIP. As ferramentas escolhidas para dar suporte ao desenvolvimento foram Visual Studio Code na versão 1.20.1, XAMPP na versão 5.6.20-0 e PhpMyAdmin para armazenar o banco de dados.

Após a definição das características técnicas do sistema, foi elaborado um diagrama Entidade-Relacionamento para modelar a base de dados, com auxílio da ferramenta Lucid-chart. No diagrama, as abreviações pk e fk referem-se à chave primária e chave estrangeira, respectivamente.

Figura 5 – Diagrama Entidade-Relacionamento do SIP.



Fonte: Próprio autor.

A documentação do sistema foi realizada de forma simples e concisa, consequência da utilização do Scrum no processo de desenvolvimento do SIP. Os relatórios de desempenho e de tarefas realizadas foram organizados em tabelas e divididos por dia de trabalho. A cada dia, tarefas eram delegadas e o desenvolvedor relatava o sucesso ou não em realizá-las e suas dificuldades, caso houvessem.

Figura 6 – Documentação de Tarefas.

		SPRINT #1 DAILY SCRUM		
DURAÇÃO DA SPRINT: 07 dias (dias/semanas)				
DATA	Nome	O que fiz ontem	O que vou fazer hoje	Existe Algum Impedimento?
31/08/2016 DIA 1	Gabriel Rufino	Levantamento de Requisitos	Levantamento de Requisitos	
		Levantamento de Requisitos	Levantamento de Requisitos	
		Index.php	Banco de Dados	
			Classe Protocolo	
			Sistema_prograd.sql	
01/09/2016 DIA 2	Gabriel Rufino		protocolo.adicionar.php	
		Banco de Dados	protocolo.editar.php	
		Classe Protocolo		
		Sistema_prograd.sql		
		protocolo.adicionar.php		
05/09/2016 DIA 3	Gabriel Rufino		protocolo.lista.php	
			botar views para funcionar	
			controler usuario.php	
			modelo usuario.php	
			usuario.adicionar.php	
06/09/2016 DIA 4	Gabriel Rufino		usuario.editar.php	
		protocolo.lista.php	colocar bd para funcionar	
		botar views para funcionar	criar entidade tipo de documento	
		controler usuario.php	campo para anexar arquivo	
		modelo usuario.php		
08/09/2016 DIA 5	Gabriel Rufino		colocar bd para funcionar	conhecimento aprofundado em bd
		criar entidade tipo de documento	imagem index	
		colocar bd para funcionar	fazer dropdown protocolo destinatário	
		campo para anexar arquivo	botões index	
09/09/2016 DIA 6	Gabriel Rufino		colocar bd para funcionar	conhecimento aprofundado em bd
		colocar bd para funcionar	colocar data fixa envio	
		fazer dropdown protocolo destinatário	dropdown protocolo funcionar	
		imagem index	justificar campos cadastro protocolo	
		botões index	footbar cor	
12/09/2016 DIA 7	Gabriel Rufino			
		dropdown protocolo funcionar	colocar bd para funcionar	conhecimento aprofundado em bd
		justificar campos cadastro protocolo	consertar protocolo.editar.php	
		colocar data fixa envio		
		footbar cor		

Fonte: Próprio autor.



Apesar do projeto conter apenas um desenvolvedor, dificultando assim a realização de tarefas mais complexas, que exigiam mais tempo, o sistema encontra-se funcional e com a maioria dos requisitos implementados. O tempo gasto para a implementação do SIP foi de, aproximadamente, 340 horas. A interface é composta por uma barra superior, onde o usuário tem acesso ao cadastro e gerenciamento de seus protocolos e reuniões. As telas, representadas pelas figuras abaixo, mostram a página inicial, o cadastro e lista de protocolos, respectivamente.

Figura 7 – Tela página inicial do SIP.



Fonte: Próprio autor.

Figura 8 – Tela cadastro de protocolo do SIP.

Fonte: Próprio autor.

Figura 9 – Tela lista de protocolos do SIP.

SIP - Sistema de Protocolos HOME CADASTRO GERENCIAMENTO SAIR

[Home](#) / [Gerenciamento](#) / [Protocolo](#) / [Meus Protocolos](#)

### Meus Protocolos

[Atualizar](#) [+ Adicionar Dados](#)

#	Remetente	Destinatário	Descrição do Teor	Data de Envio		
1	Reitoria	Pró-Reitor	PIS	2018-01-16	<a href="#">Editar</a>	<a href="#">Deletar</a>
2	PROACE	DAA	Monitoria	2018-03-07	<a href="#">Editar</a>	<a href="#">Deletar</a>

Fonte: Próprio autor.

## 8 CONCLUSÃO

Em um projeto de software, é necessário compreender bem as funcionalidades e a complexidade do sistema a ser desenvolvido, para que possam ser aplicadas as melhores práticas de estimação no processo de desenvolvimento. As métricas abordadas neste trabalho foram Planning Poker e Análise de Pontos de Função, sendo que ambas apresentaram pontos positivos e negativos no que tange à aplicação no desenvolvimento do SIP.

A aplicação da APF mostrou-se imprecisa sem a utilização de uma fonte adequada para a conversão de valores em pontos de função para horas de trabalho. Isso se deve ao fator de subjetividade presente no desenvolvimento de software, o que não foi levado em consideração na conversão de valores. A superestimação dos valores na APF demonstra a dificuldade em se estimar esforço necessário para o desenvolvimento de software, levando em consideração fontes de informação que não estão contextualizadas ao ambiente de desenvolvimento. Devido às dificuldades mencionadas acima, foi sugerida uma metodologia para criar a própria tabela de conversão de esforço empregado a cada ponto de função, utilizando intervalo de confiança. O uso da metodologia mostrou-se bastante precisa, pelo fato dos valores encontrados serem bastante parecidos com os gerados pelo PP e os valores reais obtidos.

O PP revelou-se uma técnica de estimativa de esforço muito pertinente ao projeto do SIP. Além da estimativa de esforço e prioridade resultantes serem condizentes com a realidade e complexidade do sistema, a técnica proporcionou interações e trocas de ideias entre os participantes da reunião (com experiência em desenvolvimento de software) e os *Product Owners*, o que agregou muito valor à estimativa e às funcionalidades do SIP. Em adição, a técnica de PP proporcionou uma maior adequação às diretrizes da metodologia ágil Scrum, utilizada no projeto.

Com a realização do estudo, foi possível concluir que ambas as métricas estudadas podem ser aplicadas a ambientes de trabalho mais informais, como por exemplo, em desenvolvimento de software de baixa complexidade. Entretanto, é importante que a APF seja adaptada ao ambiente de desenvolvimento em questão, a fim de promover maior compatibilidade com a realidade do mesmo.



## 9 CONSIDERAÇÕES FINAIS

Os resultados da aplicação da Análise de Pontos de Função mostraram-se, inicialmente, muito discrepantes em relação à aplicação do Planning Poker. Este fato ocorreu devido a utilização da tabela da [ANCINE \(2008\)](#) como parâmetro para geração das estimativas. Como a aplicação do Planning Poker contou com a participação de pessoas com experiência em desenvolvimento de software, percebeu-se que os resultados da aplicação deste estavam mais condizentes com a realidade do ambiente de desenvolvimento do SIP.

Com o intuito de aprimorar o valor das estimativas geradas pela Análise de Pontos de Função, sugeriu-se a aplicação do Intervalo de Confiança às estimativas de esforço. Ao aplica-lo, foi perceptível que os resultados obtidos mostraram-se bem parecidos com os gerados pela aplicação do Planning Poker, aumentando assim a confiabilidade dos resultados gerados pela Análise de Pontos de Função.

De forma geral, a APF é uma técnica que pode auxiliar na estimação de esforço para o desenvolvimento de software. Entretanto, deve-se agregar à mesma uma certa subjetividade, ou seja, adaptá-la ao ambiente de desenvolvimento em que será aplicada, buscando assim resultados mais condizentes com a realidade do projeto.

A aplicação do Planning Poker ocorreu em três momentos. A primeira aplicação gerou estimativas subestimadas, sendo assim, tarefas foram derivadas das histórias de usuário para a geração de estimativas mais precisas. Concluiu-se que a estimação de esforço com tarefas como parâmetro torna os valores da estimativa mais precisos. A aplicação do Planning Poker, estimando prioridade, mostrou-se bastante eficaz devido à facilidade com que os participantes da reunião trabalharam para gerar as estimativas. Foi perceptível também a importância das estimativas de prioridade para a ordem em que as tarefas devem ser desenvolvidas, possibilitando a otimização do processo de desenvolvimento e gerando maior qualidade para o sistema.

Em trabalhos futuros, seria interessante a aplicação das métricas APF e PP no desenvolvimento de sistemas mais complexos, sendo que o intervalo de confiança poderia ser aplicado à APF para geração de tabelas de consultas de esforço (em horas) para cada linguagem de programação.

Diante dos resultados obtidos durante a realização deste trabalho, concluiu-se que as métricas de software devem ser aplicadas de forma adaptativa, ou seja, levando em consideração as características individuais de cada projeto.



## REFERÊNCIAS

- ACTION, P. **Distribuição T de Student**. 2018. Disponível em: <http://www.portalaction.com.br/probabilidades/64-distribuicao-t-de-student>.
- ANCINE. **Anexo XVIII - Tabelas de Produtividade Mínima**. 2008. Disponível em: <http://www.ancine.gov.br/media/concorrencia0012008/AnexoXVIII.pdf>.
- BISSI, W. Scrum - metodologia de desenvolvimento Ágil. **Campo Digital**, 2007. Disponível em: <http://revistas.bvs-vet.org.br/campodigital/article/view/30944/33947>.
- CORDEIRO, M. A. Blog, **Métricas de Software**. 2018.
- CORREA, S. M. B. B. **Probabilidade e Estatística**. 2. ed. [S.l.]: PUC Minas Virtual, 2003.
- DEKKERS, C. A. Pontos de função e medidas: O que é um ponto de função. **Tradução de Mauricio Aguiar**, 1999. Disponível em: [http://www.academia.edu/download/38097381/Pontos\\_de\\_Funcao\\_e\\_Medidas.doc](http://www.academia.edu/download/38097381/Pontos_de_Funcao_e_Medidas.doc).
- DIMITRIJEVIC, S.; JOVANOVIC, J.; DEVEDZIC, V. A comparative study of software tools for user story management. **Information and Software Technology**, 2014. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950584914001293>.
- GAMBA, M. L.; BARBOSA, A. C. G. Engenharia de Software-Aplicação de Métricas de Software com Scrum. **Anais SULCOMP**, v. 5, 2010. Disponível em: <http://periodicos.unesc.net/sulcomp/article/download/242/247>.
- HAUGEN, N. C. An empirical study of using planning poker for user story estimation. In: **Agile Conference, 2006**. IEEE, 2006. p. 9–pp. Disponível em: <http://ieeexplore.ieee.org/abstract/document/1667560/>.
- JUNIOR, M. de F.; FANTINATO, M.; SUN, V. Improvements to the function point analysis method: A systematic literature review. **IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT**, 2015. Disponível em: <http://ieeexplore.ieee.org/abstract/document/7165621/>.
- LOPES, J. d. S.; BRAGA, J. L. **Guia Prático em Análise de Ponto de Função**. 2011.
- MOLOKKEN-OSTVOLD, K.; HAUGEN, N. C. Combining estimates with planning poker—an empirical study. In: **Software Engineering Conference, 2007. ASWEC 2007. 18th Australian**. IEEE, 2007. p. 349–358. Disponível em: <http://ieeexplore.ieee.org/abstract/document/4159687/>.
- PATINO, C. M.; FERREIRA, J. C. Intervalos de confiança: uma ferramenta útil para estimar o tamanho do efeito no mundo real. **J Bras Pneumol**, 2015. Disponível em: [http://www.scielo.br/pdf/jbpneu/v41n6/pt\\_1806-3713-jbpneu-41-06-00565.pdf](http://www.scielo.br/pdf/jbpneu/v41n6/pt_1806-3713-jbpneu-41-06-00565.pdf).
- PRIMO, G. Blog, **User Stories – O que são? Como Usar?** 2011. Disponível em: <http://blog.myscrumhalf.com/2011/10/user-stories-o-que-sao-como-usar/>.
- REGO, M. F.; BATISTA, G.; RIBEIRO, E. da S. Estudo de caso 02: Comparação do imc médio dos alunos do ppgee-ufmg ao longo de um ano. 2017.

SANTIN, L. D. C. ANÁLISE DE PONTOS DE FUNÇÃO: UM ESTUDO DE CASO EM UMA EMPRESA COM MPS. BR NÍVEL F. 2014. Disponível em: <http://www.uel.br/cce/dc/wp-content/uploads/TCC-LeticiaSantin-BCC-UEL-2014.pdf>.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum**. 2016. Disponível em: <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Portuguese-Brazilian.pdf>.

SYNERGIA. **Definindo a prioridade das histórias utilizando o quadro Esforço x Valor**. 2015. Disponível em: <http://www.synergia.dcc.ufmg.br/definindo-prioridade-das-historias-utilizando-o-quadro-esforco-x-valor/>.

TAMRAKAR, R.; JØRGENSEN, M. Does the Use of Fibonacci Numbers in Planning Poker Affect Effort Estimates? **IET ISBN 978-1-84919-541-6**, 2012. OCLC: 812609125.

TORRECILLA-SALINAS, C.; SEDEÑO, J.; ESCALONA, M.; MEJÍAS, M. Estimating, planning and managing Agile Web development projects under a value-based perspective. **Information and Software Technology**, v. 61, p. 124–144, maio 2015. ISSN 09505849. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0950584915000142>.

UNB. **Tabela T de Student**. 2018. Disponível em: <https://aprender.ead.unb.br/mod/resource/view.php?id=81309>.

VAZQUEZ, C. E.; SIMOES, G. S.; ALBERT, R. M. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software**. 9. ed. [S.l.]: Editora Érica, 2010.

VENTURA, P. **Valor do Ponto de Função**. 2015. Disponível em: <http://www.ateomomento.com.br/valor-do-ponto-de-funcao/>.