

CRIZON 2.0: UMA FERRAMENTA PARA ESTIMATIVA DE PONTOS DE FUNÇÃO USANDO DIAGRAMA DE SEQUÊNCIA DA UML E A INTERPRETAÇÃO DO SCRIPT DDL DA SQL

Djonathas C. Cardoso¹
Sérgio S. Barros¹
Victor E. S. Silva¹
Arlenes B. Delabary²

RESUMO

Para facilitar a determinação de um valor real para o software, Allan Albrecht (1979) elaborou a técnica Análise de Pontos de Função, cujo objetivo consiste em mensurar o valor funcional de um software independentemente das tecnologias utilizadas na implementação. A partir dessa técnica, a equipe ou gerente do projeto é capaz de determinar a quantidade de Pontos de Função de um projeto, assim, determinando um valor real do custo de desenvolvimento do software. Neste contexto, o presente trabalho discorre sobre o desenvolvimento de uma ferramenta para estimativa de Pontos de Função usando diagrama de sequência da UML e a interpretação do *script* DDL da SQL; tendo como objetivo auxiliar e facilitar o processo de análise e cálculo de Pontos de Função, de forma que seja mitigado o tempo da análise e eliminando a necessidade de possuir um conhecimento abrangente sobre a técnica de Pontos de Função. Os resultados obtidos demonstram um ganho de tempo na realização da análise a partir da ferramenta, além de mostrar que o fluxo de uma funcionalidade é mais visível na construção do diagrama de sequência, levando a uma análise mais concisa.

Palavras-Chave: Análise de Pontos de Função. Métrica de *software*. Ferramenta de Métrica de *Software*.

ABSTRACT:

To facilitate the determination of a real value for the software, Allan Albrecht (1979) elaborated the technique analysis of function points, whose objective consists of measuring the functional value of a software independently of the technologies used in the implementation. From this technique, the team or project manager is able to determine the amount of function points of a project, thereby determining a real value of the software development cost. In this context, this work discusses the development of a tool for estimating function points using UML's sequence diagram and the interpretation of the SQL DDL script; Aiming to help and facilitate the process of analyzing and calculating function points, so that the time of analysis is mitigated and eliminating the need to possess comprehensive knowledge of the technique of function points. The results obtained demonstrate a time gain in performing the analysis from the tool, in addition to showing that the flow of a functionality is more visible in the construction of the sequence diagram, leading to a more concise analysis.

¹ Acadêmicos do curso de Especialização Lato Sensu em Desenvolvimento de *Software* para Dispositivos Móveis – Faculdade Católica do Tocantins (FACTO). E-mail: djonathas@gmail.com, sergyobarrus@gmail.com, victoreduardo.s.s@gmail.com.

² Mestre em Educação. Professora na Faculdade Católica do Tocantins (FACTO). E-mail: arlenes.delabary@catolica-to.edu.br.

Keywords: Analysis of function points. Software metrics. Software metric tool.

1. INTRODUÇÃO

A gerência de projetos tornou-se de suma importância no desenvolvimento de *software*, pois proporcionou maior organização de prazo, custo, tempo e manutenção da qualidade do projeto. Segundo uma das maiores associações de gestores de *software*, a *Project Management Institute* – PMI, “o gerenciamento de projetos é a aplicação de conhecimentos, habilidades e técnicas para a execução do projeto de forma efetiva e eficaz” (PMI, 2014, s/p). Para que essa união entre conhecimento e habilidades possibilitem um bom gerenciamento, é necessário que o gestor se mantenha sempre atualizado e busque aprimoramento, amadurecimento e aplicação de atividades relacionadas à gerência, pois estas atividades influenciam diretamente na qualidade, e certamente, no sucesso de projetos de *software*.

O gerente de projeto, que tem como papel coordenar, monitorar e gerenciar o desenvolvimento do projeto, precisa possuir habilidades e conhecimentos que possibilitem um bom gerenciamento dos projetos. Além disso, ele tem o papel de mensurar o tamanho do *software* a ser desenvolvido, com o intuito de determinar um valor quantitativo a ser cobrado para desenvolver o *software*. Esse valor é obtido a partir de quatro fatores: o tamanho do produto a ser desenvolvido, o esforço a ser empregado para sua implementação, a duração do projeto e o custo gerado pela organização para o desenvolvimento.

Para facilitar a determinação de um valor real para o *software*, Allan Albrecht, em 1979, elaborou a técnica Análise de Pontos de Função – APF (IFPUG, [s.d.]), que visa mensurar o valor funcional de um software independentemente das tecnologias utilizadas na implementação. A partir dessa técnica, a equipe ou gerente do projeto tem a possibilidade de determinar a quantidade de Pontos de Função de um projeto, assim, pode-se obter um valor real do custo de desenvolvimento do *software*.

A técnica divide o *software* em dois tipos de análise, que são: Grupo de Dados, que representa a estrutura da base de dados; e Transações, que consistem no mapeamento das funcionalidades ou ações realizadas pelo *software*. Segundo o autor, cada etapa, ao final, representará um valor em Pontos de Função, que será somado para definir o tamanho funcional do *software* como um todo.



O problema relacionado a esta técnica é que sua aplicação envolve tarefas manuais, necessitando sempre um grande gasto de tempo para realizar a análise funcional do *software* a partir da visão do usuário. Esse gasto de tempo reflete na análise de visões diferentes do *software* na qual, primeiramente, é analisada a estrutura dos dados presentes e, posteriormente, as funcionalidades ou transações existentes, levando sempre em consideração a visão do usuário.

Assim, para facilitar a utilização da técnica e torná-la mais automática, o presente trabalho discorre sobre o desenvolvimento de três módulos de um sistema que possibilita a mensuração do tamanho funcional dos *softwares* a partir da técnica de análise de Pontos de Função, utilizando como entrada os Diagramas de Sequências e o *script Data Definition Language* - DDL do banco de dados. Estes dois artefatos foram escolhidos por possibilitar a identificação dos Grupos de Dados e das Transações envolvidas no *software*.

Por fim, essas funcionalidades foram unidas à ferramenta CRIZON, desenvolvida por Rodrigues (2012), que possibilita a criação de artefatos de modelagem para *software* e, com as novas funcionalidades, possibilita também a mensuração do tamanho funcional a partir da análise do *script* DDL do banco de dados e diagramas de sequência construídos pelo usuário.

2. ANÁLISE DE PONTOS DE FUNÇÃO

Esta técnica busca criar uma unidade de medida de *software*, como é o caso do m², que é uma unidade de medida de comprimento utilizada para, por exemplo, determinar o tamanho de um terreno. Neste caso, o valor para adquirir o referido terreno é diretamente relacionado ao tamanho do mesmo. Dessa forma, a unidade de medida da APF, denominada Pontos de Função, seria utilizada para mensurar o tamanho funcional do *software*, de acordo com a visão do usuário, assim, determinando a partir da quantidade de PF o valor real do *software*.

Dessa forma, a técnica de APF consiste em duas fases, são elas: a primeira fase é composta pelo processo de determinação da complexidade de dados e transações contidas no *software* em análise; e a segunda fase é composta pela realização do cálculo e determinação da quantidade de pontos de função que possui o *software* em análise.

O Processo de Determinação é a fase do processo de contagem de pontos de função responsável por realizar inferências e determinar a complexidade dos dados e transações contidas no sistema sob análise. Esse processo agrupa duas etapas de grande importância na contagem, são elas: Complexidade de Dados, que representa a etapa de análise dos dados utilizados no sistema, a partir da visão do usuário; e, a Complexidade de Transação, que é a etapa de análise e levantamento das operações realizadas no sistema.

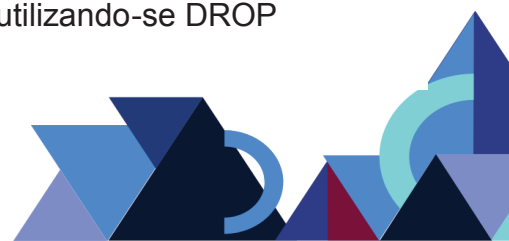
Já o Processo de Contagem ocorre somente quando a complexidade do grupo de dados e transações foi definida a partir do processo de determinação da seção anterior. Com isso, para determinar a quantidade de Pontos de Função que o *software* em análise contém, é preciso somar a quantidade de pontos de função de cada grupo de dado e transação.

Na seção 3, a Linguagem de Definição de Dados (DDL) da SQL será apresentada, bem como os principais conceitos do Diagrama de Sequência da UML, que também serão utilizados como entrada da contagem de Pontos de Função

3. DATA DEFINITION LANGUAGE (DDL) DA SQL

A Linguagem de Definição de Dados, em inglês, *Data Definition Language*, é composta pelos “comandos para definição das estruturas das tabelas e dos tipos de atributos respectivos” (MEDEIROS, 2013, p. 96), mais conhecidos como os comandos *CREATE*, *DROP* e *ALTER*. Cada comando tem uma definição e função distinta, sendo elas:

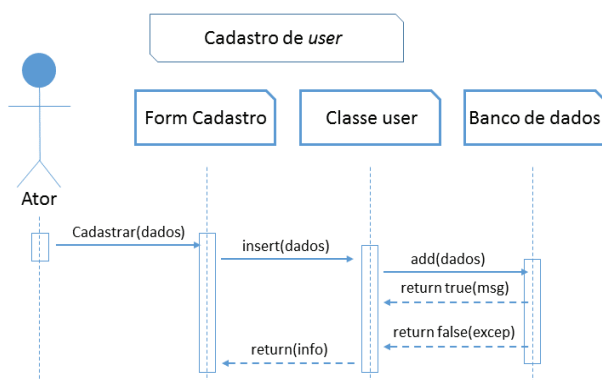
- Instrução *Create Table*: responsável por definir a estrutura dos objetos do banco de dados, contendo respectivamente o nome da tabela, colunas, tipos de dado de cada coluna e informações complementares.
- Instrução *Alter Table* – “modifica uma definição de tabela alterando, adicionando ou removendo colunas e restrições, reatribuindo e recriando partições, ou desabilitando ou habilitando restrições e gatilhos” (MSDN, [s.d.]).
- Instrução *Drop Table* – “remove uma ou mais definições de tabela e todos os dados, índices, gatilhos, restrições e especificações de permissão dessas tabelas”. Qualquer exibição ou procedimento armazenado que faça referência à tabela descartada deverá ser descartado explicitamente utilizando-se *DROP VIEW* ou *DROP PROCEDURE* (MSDN online,[s.d]).



4. DIAGRAMA DE SEQUÊNCIA DA UML

Segundo Guedes (2011, p. 193), este é um diagrama comportamental que procura determinar a sequência de eventos que ocorrem em um determinado processo, identificando quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem. Assim, esse diagrama busca relatar quais passos são necessários para realizar uma determinada ação. Esses passos vão desde identificar quais classes serão utilizadas para realizar a ação, até as mensagens e dados de retorno ao final de cada interação. A Figura 1 ilustra um exemplo do diagrama de sequência.

Figura 1 - Exemplo de Diagrama de Sequência da UML



Fonte: Acervo dos autores

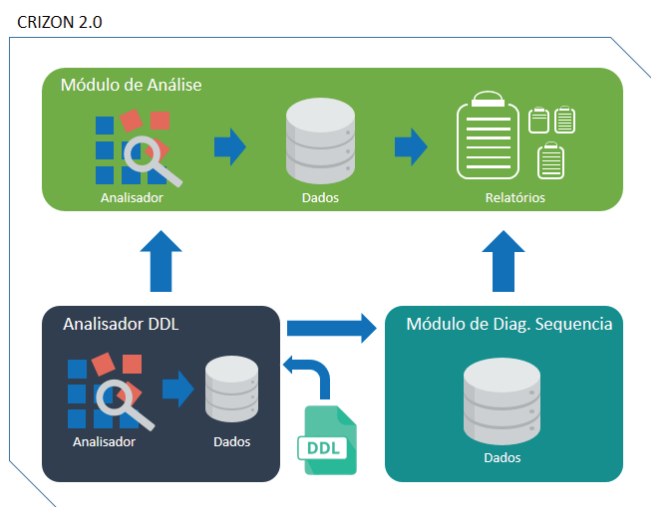
O diagrama “é organizado colocando-se os objetos correspondentes na parte superior, ao longo do eixo horizontal; e suas respectivas mensagens são colocadas ao longo do eixo vertical, em uma ordem cronológica, de cima para baixo” (SEABRA, 2001, p.17). Assim, o diagrama de sequência descreve em uma ordem cronológica os passos que serão realizados para que determinada funcionalidade seja completada, ilustrando a comunicação entre o ator e objetos.

5. METODOLOGIA

Para que seja possível realizar a análise de pontos de função a partir do diagrama de sequência da UML e *script* DDL, foi preciso, primeiramente, planejar como tais entradas iriam fornecer as informações necessárias para que a análise fosse feita com êxito. Para isso, foram construídos módulos para que as informações

tanto do diagrama, quanto do *script* fossem extraídas de maneira correta e exata, fazendo com que a análise expressasse a real visão funcional do *software*. A seguir, é apresentada, na Figura 2, a Arquitetura do *Software*, que ilustra os elementos presentes no *software*, suas interações e padrões presentes do sistema.

Figura 2 - Arquitetura dos módulos propostos a partir da visão de desenvolvimento/estrutural.



Fonte: Acervo dos autores

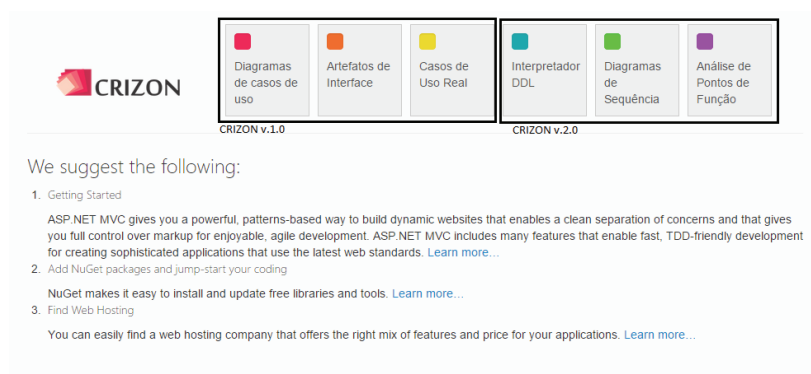
Na Figura 2, é apresentada a arquitetura estrutural dos módulos, que é dividida em três partes, sendo elas:

- **Analisador DDL:** módulo responsável por interpretar o *script* DDL da SQL, documento de entrada fornecido pelo usuário;
- **Módulo de Diagrama de Sequência:** representa o módulo que possibilita o usuário construir os diagramas de sequência que pertencem ao *software* a ser analisado para que, posteriormente, seja utilizado como entrada para análise de Pontos de Função; e,
- **Módulo de Análise** – módulo referente à análise de Pontos de Função, tendo como fonte de entrada as informações providas dos módulos anteriores que serão utilizadas para a realização da análises.

A análise realizada a partir dos módulos divide-se em três etapas, que representam desde a interpretação do *script* DDL do banco de dados até a análise dos dados existentes no sistema para inferência dos Pontos de Função.

A Figura 3 ilustra os módulos presentes no CRIZON versão 1.0 e os três módulos desenvolvidos neste trabalho, que representa a versão 2.0 do *software*.

Figura 3 - Módulos do CRIZON na versão 1.0 e 2.0



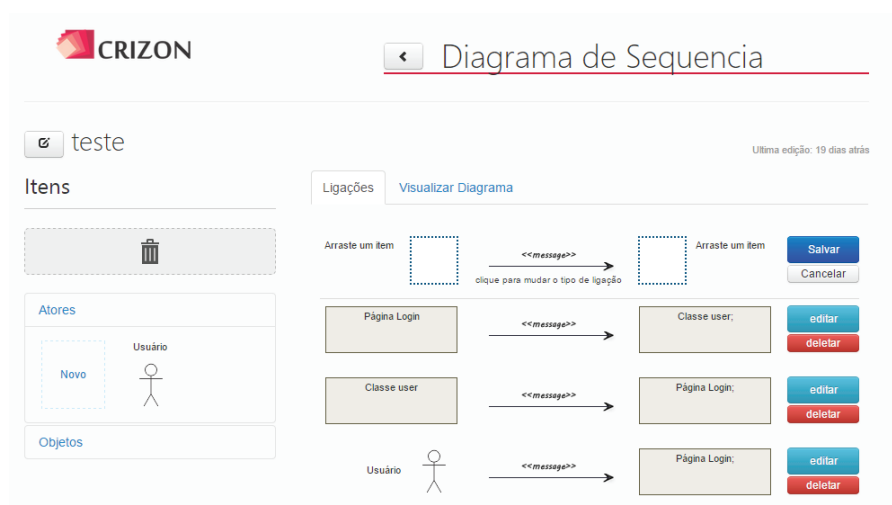
Fonte: Acervo dos autores

Os três módulos, representado pela figura acima, na versão 2.0, possibilitam a mensuração do tamanho funcional do *software* a partir de três etapas, a saber:

- Interpretação do script DDL: consiste na interpretação do script DDL do banco de dados para definição das entidades existentes no *software* a ser analisado.
- Construção dos Diagramas de Sequências: consiste na criação e definição das transações ou funções realizadas pelo *software* a ser analisado.
- Análise de Pontos de Função: representa a análise dos grupos de dados (entidades) e transações (diagramas) analisados/cadastrados de forma que indique, primeiramente, quais as entidades e diagramas participaram desta análise, bem como os tipos de dados existentes para que, ao final desta análise, seja possível inferir a quantidade de Pontos de Função que o *software* possui e gerar um relatório com as informações providas desta análise.

Dentre os três módulos desenvolvidos, o módulo construtor de diagrama de sequência é um dos módulos mais importantes, no que se refere a precisão de êxito do *software*, pois o usuário precisa construí-lo de forma idêntica à real funcionalidade relatada, para que o resultado da interpretação deste diagrama mensure o tamanho exato desta ação. Caso haja uma construção mal realizada ou o esquecimento de alguma funcionalidade, a mensuração não refletirá o tamanho exato do *software* analisado. A Figura 4 ilustra a tela de construção do diagrama de sequência.

Figura 4 - Tela de edição do diagrama de sequência



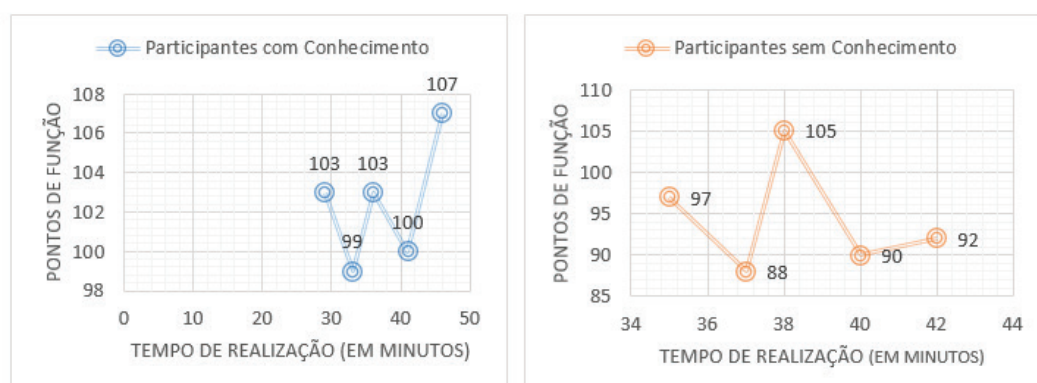
Fonte: RODRIGUES, 2012

A tela de edição, ilustrada pela Figura 4, possibilita a criação de atores que participam das ações existentes; dos objetos, na qual transitam as mensagens da ação; e ligações, que representam o elo de comunicação entre um ator-objeto. Além disso, é possível visualizar o diagrama a qualquer momento, sua atualização é simultânea de acordo com a criação das ligações existentes, e pode ser acessada na aba “Visualizar Diagrama”. Essa funcionalidade só foi possível a partir da integração com a *Application Programming Interface - API do Web Sequence Diagrams*, sistema que possibilita a criação do diagrama de sequência em sua plataforma, além de disponibilizar um *webservice* para a construção de diagramas a partir de suas definições, retornando assim uma imagem de acordo com as ligações enviadas via requisição.

6. RESULTADOS E DISCUSSÕES

Com a conclusão do desenvolvimento da ferramenta, foi realizado teste de verificação com o objetivo de averiguar a eficiência da ferramenta no que se refere à aferição da análise de pontos de função manual e automatizada pelo software. O teste foi aplicado em 10 participantes e foi possível determinar algumas visões, bem como um comparativo entre as análises realizadas manualmente e automatizada. A visão a seguir, exhibe a relação entre pontos de função e tempo de realização da análise a partir do comparativo entre os participantes com e sem conhecimento sobre a técnica de Pontos de Função, como apresentado na Figura 5.

Figura 5 - Comparativo participantes com e sem conhecimento sobre a técnica a partir da relação entre pontos de função e tempo de realização da análise



Fonte: Acervo pessoal

É possível observar, a partir da Figura 5, a variação entre os pontos de função das análises realizadas entre dois grupos: participantes com e sem conhecimento. Utilizando a variante **Pontos de Função**, entre os dois grupos, a comparação apresenta uma pequena variação entre os resultados obtidos pelos participantes com conhecimento que, comparado com os participantes sem conhecimento, possui uma variação maior. Já utilizando a variante **Tempo de Realização**, também é constatada pouca variação entre os participantes com conhecimento, mas entre os participantes sem conhecimento obteve-se uma variação maior.

Entretanto, essa variação referente ao tempo comprova que os participantes com conhecimento sobre a técnica de Pontos de Função conseguem realizar a mensuração no *software* mais rapidamente, por possuir um conhecimento prévio.

Na variação entre os pontos de função, percebe-se que os tempos variam, pois cada participante tem uma visão distinta ao realizar a análise de um *software*, possibilitando assim uma variação entre os Pontos de Função resultantes. Essa variação de PF não determina que as análises estão incorretas, somente indica que cada usuário tem uma visão diferente do mesmo *software*, ou seja, como não houve nenhuma análise com uma grande variação, todas as análises realizadas estão de acordo com o *software* proposto.

É importante destacar que, ao realizar a análise pela ferramenta, o usuário constrói o diagrama de sequência, o que faz com que exista uma variação entre as funções analisadas manualmente e no sistema, pois construindo o diagrama é possível entender e visualizar a funcionalidade como ela realmente é, e não imaginar

como seria a funcionalidade a partir de uma descrição, como ocorre na contagem manual.

7. CONSIDERAÇÕES FINAIS

A determinação do tamanho de um *software* ou o valor real que será cobrado pelas funcionalidades existentes é um processo difícil, pois é preciso trazer à tona fatores que influenciam diretamente no custo do desenvolvimento como, por exemplo: mão de obra, planejamento, modelagem, testes, riscos, dentre outros fatores.

O objetivo desta ferramenta foi automatizar o processo que, anteriormente, era realizado de modo manual, necessitando muito tempo para sua conclusão. Outro problema com relação ao processo manual era que, a cada alteração, precisava-se realizar toda a contagem novamente. Agora, com a ferramenta, a análise pode ser alterada e recalculada em poucos cliques e quantas vezes for necessário. Além disso, é realizada a partir de um artefato de modelagem de *software* e da análise da estrutura do banco de dados. Ou seja, o processo que anteriormente era manual evoluiu para um processo automatizado que, a partir de artefatos de modelagem que já são desenvolvidos pela equipe para documentar o desenvolvimento do *software*, facilita a mensuração e elimina custos e tempo do processo de análise manual.

Algumas diferenças podem ser observadas em relação às outras ferramentas citadas nos trabalhos correlatos como, por exemplo, a utilização de um interpretador do *script* DDL para determinar os Pontos de Função dos Grupos de Dados, comparando com as ferramentas de Batista et al. (2011) e Uemura et al. (2001) que necessitam que o usuário construa modelos e/ou diagramas para realizar essa determinação. Outra diferença é que o processo de análise de Pontos de Função realizado pela ferramenta desenvolvida neste trabalho não determina a necessidade da utilização de nenhuma metodologia, como é o caso das ferramentas de Pinel (2012) e Fraternali et al. (2006), que determinam que o especialista em Pontos de Função necessita utilizar uma metodologia de desenvolvimento escolhida pelos autores.

Em contrapartida, um dos benefícios que pode ser apontado com essa ação é o tempo gasto na correção de um erro, na qual esse tempo na ferramenta é menor em comparação com a fase manual, pois não necessita realizar a contagem e analisar



a funcionalidade novamente, já que, no sistema, o diagrama pode ser alterado quando for necessário e recalculado a qualquer momento. Além disso, ao construir os diagramas, o usuário tem a visão de como realmente executará a funcionalidade a partir das comunicações entre ator/objeto. Já na fase manual, essa visão não acontece e o usuário precisará recorrer ao papel para descrever uma ideia, que talvez não refletirá a funcionalidade real da ação. Ou seja, no processo manual o usuário não tem aparatos para determinar as transações e precisa buscar meios que ajudem a determiná-las. Já no sistema, as funcionalidades são analisadas pelo diagrama de sequência construído.

Por fim, os resultados obtidos na fase de validação foram satisfatórios, pois comprovaram um ganho de tempo na realização da análise a partir da ferramenta, além de demonstrar que o fluxo de uma funcionalidade é mais visível na construção do diagrama de sequência, levando a uma análise mais concisa. Além disso, os participantes também se expuseram ao descrever a facilidade da construção e a não necessidade de conhecimento, no qual influenciou consideravelmente na facilidade de utilizar a ferramenta.

REFERÊNCIA

BATISTA, Vitor A.; PEIXOTO, Daniela. C. C.; BORGES, Eduardo P.; PÁDUA, Wilson de; RESENDE, Rodolfo F.; PÁDUA, Clarindo Isaías P. S. **REMOFP: A Tool For Counting Function Points From UML Requirement Models**. *Advances in Software Engineering*, vol. 2011, p. 1-7, 2011. Disponível em: <<http://downloads.hindawi.com/journals/ase/2011/495232.pdf>>. Acesso em: 5 nov. 2017.

FRATERNALI, Piero; TISI, Massimo; BONGIO, Aldo. **Automating function point analysis with model driven development**. In: *Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research*. IBM Corp., 2006. p. 18. Disponível em: <<http://www.webml.org/webml/upload/ent5/1/FP06.pdf>>. Acesso em: 6 de nov. 2017.

GUEDES, Gilleanes T. A. **UML 2: uma abordagem prática**. 2. ed. São Paulo: Novatec Editora, 2011.

MEDEIROS, Luciano Frontino de. **Banco de dados: princípios e práticas**. Curitiba: InterSaber, 2013.

MSDN, **Microsoft Developer Network**. [s.d.]. Disponível em: <[http://msdn.microsoft.com/pt-br/library/ie/6974wx4d\(v=vs.94\).aspx](http://msdn.microsoft.com/pt-br/library/ie/6974wx4d(v=vs.94).aspx)>. Acesso em: 5 nov. 017.

PINEL, Roque E. Assumpção. **Análise de Pontos de Função em Sistemas Desenvolvidos usando MDA**. Dissertação (Mestrado em Engenharia de Sistemas e Computação). 2012. Universidade Federal do Rio de Janeiro, Rio de Janeiro, 98 p. Disponível em: <http://objdig.ufrj.br/60/teses/coppe_m/RoqueEliasAssumpcaoPinel.pdf>. Acesso em: 6 nov. 2017.

PMI, **PROJECT MANAGEMENT INSTITUTE**. [s.d.]. Disponível em: <<https://brasil.pmi.org/brazil/AboutUS/WhatIsProjectManagement.aspx>>. Acesso em: 4 nov. 2017.

RODRIGUES, William Almeida. **CRIZON**: uma ferramenta para modelagem de software. Centro Universitário Luterano de Palmas, Palmas, Tocantins, 2012.

SEABRA, Rodolfo Moacir Júnior. **Análise e projeto orientado a objetos usando UML e o Processo Unificado**. Dissertação (Bacharelado em Ciência da Computação). 2001. Universidade Federal do Pará, 113 p. Disponível em: <<http://cultura.ufpa.br/cdesouza/teaching/cedai/APOOUMLP.pdf>>. Acesso em: 5 nov. 2017.

IFPUG. INTERNATIONAL FUNCTION POINT USERS GROUP. **Sobre análise de Pontos de Função**. Disponível em: <<http://www.ifpug.org/about-ifpug/about-function-point-analysis/?lang=pt>>. Acesso em: 3 nov. 2017.

UEMURA, Takuya; Kusumoto, Shinji; Inoue, Katsuro. **Function-Point analysis using design specifications based on the Unified Modelling Language**. Journal of Software Maintenance and Evolution: research and Practice, vol. 13, n. 4, p. 223-243, 31 ago. 2001. Disponível em: <http://www.researchgate.net/publication/220674140_Function-point_analysis_using_design_specifications_based_on_the_Unified_Modelling_Language/links/09e4150aa32b36a9b5000000>. Acesso em: 3 nov. 2017.

