

Sistema de medição de software baseado em pontos de função e análise baseada em técnicas de aprendizagem de máquina

Software measurement system based on function points and analysis based on machine learning techniques

Iran Alyf Albino

Graduando do curso de Sistemas de Informação (UNIPAM).

E-mail: iranalyf@unipam.edu.br

Eduardo Henrique Silva

Professor orientador (UNIPAM).

E-mail: eduardohs@unipam.edu.br

Resumo: O objetivo principal da pesquisa foi desenvolver uma plataforma web para realizar medição de softwares baseado na técnica de Análise de Pontos de Função em uma arquitetura de software baseado em *Microservices*. Pela plataforma, é possível realizar uma medição de software e, com o resultado de uma medição, é possível estimar alguns artefatos do projeto de software como tempo de desenvolvimento, custo e esforço.

Palavras-chave: *Microservices*. Análise de ponto de função. Feedback inteligente. Serviço Cognitivo.

Abstract: The main objective of the project was to develop a web platform to perform software measurement based on the technique of Function Point Analysis in a software architecture based on *Microservices*. Through the platform, it is possible to perform a software measurement and with the result of a measurement, it is possible to estimate some software design artifacts such as development time, cost and effort.

Keywords: *Microservices*. Function point analysis. Feedback intelligence. Cognitive Service.

1 INTRODUÇÃO

Nos processos de desenvolvimento de software, segundo Vazquez (2013) geralmente não é feita uma estimativa apropriada de tempo, custo e alocação de recursos necessários para um software específico. Em consequência, são bastante prováveis cenários negativos, como atraso e desperdício de recursos. Quando se têm os requisitos necessários para o processo de desenvolvimento de um software, com as técnicas apropriadas, é possível alcançar o sucesso. Em um projeto de software em andamento, também não é feito o gerenciamento de mudanças, que, caso venha acontecer, é feito muito retrabalho e há perda de tempo e recursos.

Nesse contexto, um sistema para estimar esses artefatos é de grande valia, pois fornece as informações necessárias para o sucesso do desenvolvimento de um software. Com a utilização de Análise de Pontos de Função (APF), é possível fazer a medição de um sistema completo, com base em funcionalidades do ponto de vista do usuário.

Na Inteligência Artificial, tem-se o ramo de aprendizagem de máquina, em que se podem usar alguns algoritmos para fazer análises em medições já cadastradas em uma base de dados. Com base em Aprendizado de Máquina, conseguem-se variáveis, de artefatos de medições passadas, para avaliar uma nova medição e ajudar possíveis projetos de software.

Assim foi desenvolvido um sistema de orçamento de software, que, além de favorecer os *stakeholders* a fornecer a medição, estimativas e gerenciamento do projeto, ele é capaz de gerar volumes de dados de projetos passados, a fim de auxiliar futuros projetos, passando um *feedback* de forma inteligente para o usuário do sistema.

O objetivo geral deste trabalho foi modelar e desenvolver um sistema de orçamento de software, baseado em pontos de função, utilizando dados de projetos passados, para auxiliar futuros projetos. Esse sistema possibilita a realização de estimativas de projetos de software, projeto de desenvolvimento, projeto de melhoria ou aplicação. É possível estimar tamanho, custos, tempo de desenvolvimento e consequentemente estimar quantidade de pessoas para trabalhar no projeto e ainda manter uma base de dados, de sucesso ou fracasso de projetos anteriores, para que, de forma inteligente, possa auxiliar futuros projetos. Para atingir o objetivo geral, foram desenvolvidos os seguintes objetivos específicos:

- permitir determinar o tipo de contagem, como projeto de desenvolvimento, projeto de melhoria ou aplicação;
- permitir levantar variáveis que possam contribuir para verificar os requisitos do sistema;
 - permitir ser feita a contagem de funções, como tipo dados, e tipo transações;
 - permitir ser determinado o valor do Fator de Ajuste ;
 - permitir determinar a contagem de pontos de função não ajustados;
 - permitir calcular o número dos pontos de função ajustados;
 - extrair os dados necessários para utilização dos algoritmos de aprendizado de máquina;
 - comparar o resultado de projetos anteriores, auxiliando novos projetos, baseado em aprendizado de máquina;
 - exibir gráficos apresentando informações de andamento e status do projeto;
 - disponibilizar relatórios com possíveis resultados da medição.

Atualmente, não existe um produto *open-source* igual a esse ou mesmo com vantagens parecidas no mercado. Levando em consideração que os projetos de softwares sem gerência tendem ao fracasso segundo Sommerville (2011), um produto como esse pode se tornar muito útil para pequenas, médias e grandes empresas e instituições.

2 REFERENCIAL TEÓRICO

2.1 APRENDIZAGEM DE MÁQUINA

A área de aprendizado de máquina (AM) é considerada um ramo da Inteligência Artificial. É uma área cujo objetivo é construir sistemas que sejam capazes de aprender de forma automatizada a partir de dados (BRINK; RICHARDS, 2014). Algoritmos de AM aprendem a induzir uma função ou hipótese capaz de resolver um problema a partir desses dados que representam instâncias do problema a ser resolvido (CARVALHO, 2011).

Segundo Carvalho (2011), algoritmos de AM são utilizados em diversas tarefas, que podem ser organizadas de acordo com diferentes critérios. Essas tarefas de aprendizado podem ser divididas em descritivas e preditivas. Nas tarefas descritivas, a meta é explorar ou descrever um conjunto de dados. Já nas tarefas preditivas, a meta é encontrar uma função a partir dos dados de treinamento a qual possa ser usada para prever valores desconhecidos ou futuros de outras variáveis.

No caso deste trabalho, será usada principalmente a técnica de aprendizagem indutiva, supervisionada, ou seja, a partir da base de dados existente, espera-se obter informações (modelos) preditivos, em relação às medições já realizadas.

2.2 ANÁLISE DE PONTOS DE FUNÇÃO

Segundo Vazquez (2013), a métrica de tamanho funcional Pontos de Função afere o tamanho de um projeto ou aplicação de software, levando em consideração as funcionalidades requisitadas e recebidas pelo usuário, independentemente da tecnologia e do processo de software utilizado. Com isso, é possível observar uma série de benefícios listados por Vazquez (2013):

- controlar o andamento da produtividade de um determinado software. Um sistema pode ter mais de uma equipe envolvida em seu desenvolvimento. É possível avaliar a produtividade de diferentes equipes pela quantidade de pontos de função entregados;
- realizar a medição do tamanho do software e, com isso, estimar, custo, esforço e prazo. Uma vez realizada a medição ou estimativa de ponto de funções totais do sistema, é possível utilizar esse número para realizar derivações;
- sabendo-se do o tamanho funcional de um software, é possível realizar comparações. Pode ser realizada uma avaliação entre dois ou mais sistemas;
- com a utilização da técnica, é possível tomar decisões do tipo “Make or Buy”, seria a decisão de desenvolver um sistema ou comprar uma solução pronto no mercado;
- utilizar a medida para fundamentar contratos de compra e venda de softwares ou contratar serviços.

3 METODOLOGIA

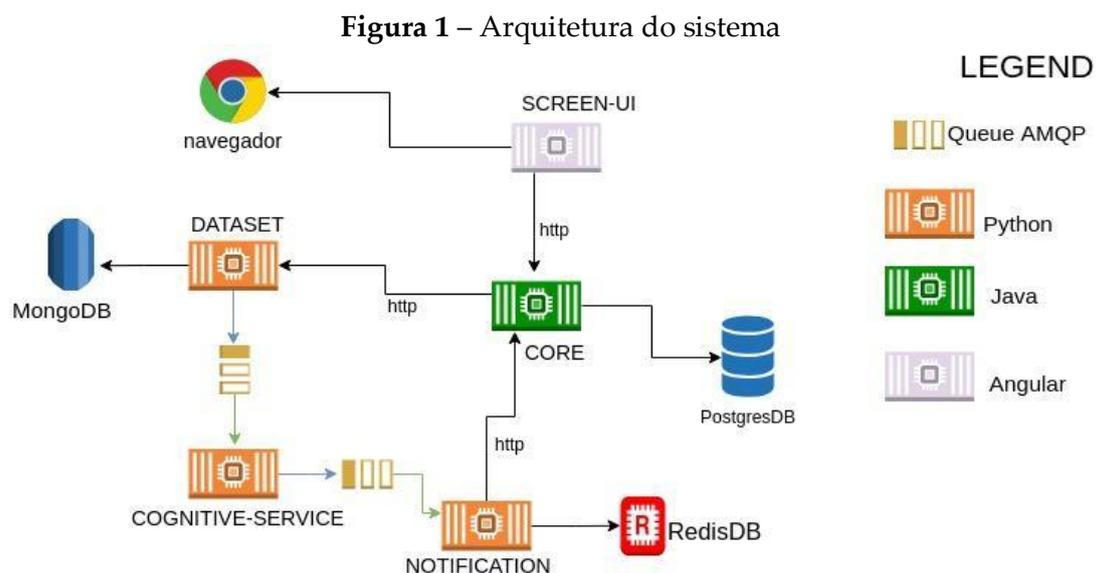
No primeiro passo, foi realizado um estudo bibliográfico sobre Análise de Pontos de Função e Técnicas de Aprendizagem de Máquina. Logo após, foi feita a modelagem do sistema, utilizando Diagramas de Entidade de Relacionamento, buscando levantar os requisitos funcionais do sistema e desenhar a Arquitetura do sistema, definindo os serviços que foram desenvolvidos e a forma como esses serviços iam se comunicar entre si.

Foi utilizada uma arquitetura de *Microservices*, ou seja, pequenos serviços com responsabilidades de funcionalidades bem específicas a cada micro serviço. No serviço de realizar a medição, em que há relacionamentos com entidades para realizar cálculos, foi utilizada a linguagem Java e o banco de dados PostgreSQL. Nos serviços em que os dados não necessariamente são estruturados, foi utilizada a linguagem Python para tratar os dados e realizar o fluxo de forma assíncrona.

4 DESENVOLVIMENTO E RESULTADOS

4.1 ARQUITETURA

Foi realizada a modelagem do sistema, construindo um Diagrama de Entidade Relacionamento (DER), conforme os requisitos levantados para o desenvolvimento do sistema. Posteriormente, foi desenhada a Arquitetura do Sistema, representada pela Figura 1.



Fonte: Elaborado pelos autores, 2018.

O serviço CORE foi desenvolvido usando a tecnologia Java na versão 10. É nesse serviço que estão implementadas as regras de negócios do fluxo de medição de software. Essas regras de negócio têm uma quantidade considerável de cálculos, para garantir a qualidade das regras. Foram implementados testes unitários com a

ferramenta JUnit para aplicações com a tecnologia Java. Os dados nesse mesmo serviço estão sendo armazenados no Banco de Dados Relacional PostgreSQL. O serviço fornece as APIs no padrão REST, que são consumidas pelo serviço SCREEN-UI, que foi desenvolvido usando a linguagem Typescript e o framework Angular na versão 6.0.

O serviço SCREEN-UI é responsável pela apresentação do sistema e interação com o usuário. Nele o usuário pode cadastrar um fluxo de medição e receber os *feedbacks* dele. Esse serviço consome APIs do serviço CORE para registrar os fluxos de medições.

O serviço DATASET é o serviço que realiza os tratamentos de dados dos usuários do sistema. Esse serviço recebe os dados de medições cadastradas no serviço CORE e organiza os dados necessários para o serviço COGNITIVE-SERVICE consumir e armazena esses dados no Banco de Dados não relacional orientado a documentos MongoDB.

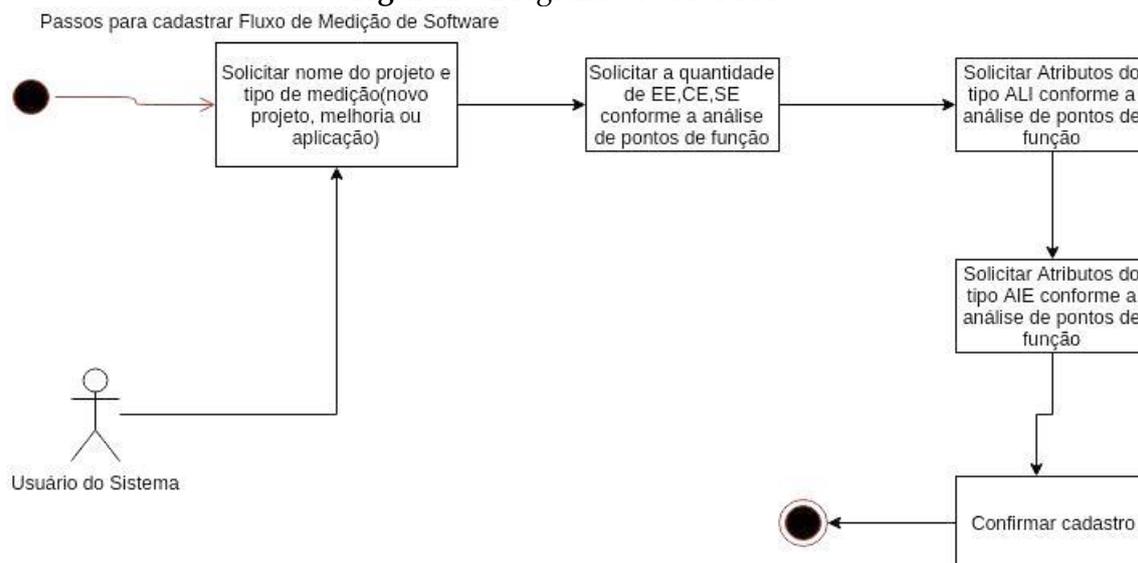
No serviço COGNITIVE-SERVICE ficam as lógicas de Inteligência do sistema e é validado se existe algum *feedback* útil com base nos projetos anteriores cadastrados por usuário. Caso existe algum *feedback* útil para o usuário que registrou um novo fluxo de medição, é enviado uma mensagem para o outro serviço NOTIFICATION.

O serviço APFI-Flow-Response é responsável para fazer uma lógica de sugestão para o usuário do sistema, caso tenha algum *feedback* processado pelo serviço COGNITIVE-SERVICE; caso tenha algum *feedback*, é enviada uma notificação para o CORE, informando o usuário que ele possui um *feedback* daquele fluxo de medição registro por ele. A resposta de *feedback* é armazenada no Banco de Dados não relacional RedisDB em cache. O usuário pode analisar o *feedback* ou ignorá-lo.

4.2 FUNCIONALIDADES

Foi desenvolvido um sistema para possibilitar realizar medições de softwares baseado nas técnicas de APF; no sistema, é possível cadastrar um cliente e esse cliente cadastrar os seus usuários. Cada usuário pode registrar fluxos de medições no sistema. Para cadastrar o fluxo de medição, são necessários os artefatos que a APF precisa, como diagrama casos de uso, diagramas UML e protótipo do sistema. Com essas informações, o usuário está apto a passar essas informações para o sistema, que realiza os cálculos necessários e irá exibir um relatório com as informações de tempo aproximado de desenvolvimento, esforço do projeto e custo baseado no valor de cada recurso da empresa. Na Figura 2, segue uma imagem explicativa de como é feito o cadastro de medição.

Figura 2 – Diagrama de atividades



Fonte: Elaborado pelos autores, 2018.

O sistema possui uma tela de relatórios em que são mostradas várias estatísticas em gráficos. É possível analisar informações das evoluções de projetos baseadas no tamanho do projeto e comparar com projetos passados.

5 CONCLUSÃO

As principais dificuldades enfrentadas foram criar uma interface de fácil manuseio para o usuário do sistema. Devido às técnicas da APF necessitarem de protótipos e diagramas, fica complicado extrair essas informações a partir desses artefatos. Nesse propósito, o usuário do sistema deve preencher uma planilha Excel com algumas informações a partir do Diagrama de UML e contar as funcionalidades do sistema a partir do protótipo e descrevê-las no sistema de medição de forma manual. Futuramente, podem ser desenvolvidas as funcionalidades para conseguir fazer o *upload* do arquivo de diagrama UML a partir do sistema, e o próprio sistema conseguir identificar as características necessárias. Futuramente também pode ser realizada a implementação de *upload* de um arquivo com um protótipo do sistema a ser medido e o próprio sistema conseguir contar as funcionalidades a partir do arquivo com o protótipo.

Foram cadastrados vários projetos de tamanhos distintos. Levando-se em consideração a quantidade de pontos de função que cada usuário consegue desenvolver, é possível estimar o tempo de desenvolvimento antes mesmo de a equipe começar a desenvolver e, em tempo de execução do projeto, serem analisadas métricas em gráficos, se aquele tempo estimado está sendo cumprido pela equipe de desenvolvedores.

Foram realizados testes utilizando IA para classificar os projetos e assim prever a estimativa em tempo dos projetos. No aprendizado supervisionado, tem-se o algoritmo KNN, que é um algoritmo simples de classificação; ele é usado para classificar objetos com base em um treinamento e, no exemplo do sistema, ele iria

classificar os projetos de acordo com as características em comuns dos projetos, como quantidade de CRUDs, que são exemplos mais simples. O problema encontrado com o algoritmo foi calcular a distância entre o exemplo desconhecido e outros exemplos do conjunto de treinamento, visto que, em nível funcionalidades de sistemas, são muito comuns funcionalidades bem distintas em relação a outros sistemas. Outra dificuldade encontrada com o algoritmo é que, para treinar o nosso modelo, é preciso que uma quantidade considerável de projetos cadastrados no sistema, ou seja, um *dataset* para se realizarem os testes necessários, e com poucos dados é inviável.

No futuro, quando houver uma base com muitos dados de projetos, será possível treinar um modelo de dados e assim utilizar a IA para classificar os projetos de forma mais inteligente e conseqüentemente fornecer uma estimativa ainda mais precisa para os usuários do sistema.

REFERÊNCIAS

BRINK, H.; RICHARDS, J. **Real world machine learning**. [S. l.]: Manning Publications C.O, 2014.

CARVALHO, André Carlos Ponce de Leon F. *et al.* **Inteligência artificial: uma abordagem de Aprendizado de Máquina**. Rio de Janeiro: LTC, 2011.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson, 2011.

VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. **Análise de pontos de função**. São Paulo: Érica, 2013.