

Proposta de Projeto de Doutoramento a Desenvolver no Âmbito do 1º Concurso para Atribuição de Bolsas de Investigação na Área de Engenharia Informática

1. Título do projeto

Título: Uma abordagem analítica à análise de plataforma de desenvolvimento *low code*

Palavras-chave: Engenharia de Software, Interação Humano-Computador, Desenvolvimento *low-code*

Referência: CEE_EI_UMINHO2

2. Instituições envolvidas

Instituição onde o doutoramento será realizado: EEUM

Outras instituições participantes no projeto de investigação: n/a

3. Equipa de Orientação

Orientador: António Nestor Ribeiro

4. Descrição do Projeto

Com esta proposta de trabalhos pretende-se automatizar o processo de análise da interação com plataformas de programação *low code*. O foco está em possibilitar uma identificação precoce de potenciais problemas de utilização nestas plataformas, que possam funcionar como barreiras à sua adoção. Dois tipos base de utilizadores deverão ser considerados: programadores e não programadores.

Ao longo do tempo tem sido possível observar um progressivo elevar do nível de abstração a que o software é desenvolvido. As plataformas de programação *low code* representam mais um passo nessa tendência, passando o software a ser modelado em vez de codificado numa linguagem de programação. Apesar da proliferação recente deste tipo de plataformas (cf. OutSystems Service Studio, Microsoft PowerApps, Appian, Mendix, Google App Maker, etc.) ainda pouco é sabido sobre quais as características que potenciam, ou limitam, a adoção das mesmas por programadores e não programadores. Com efeito, apesar de ser reconhecido que o 'pensamento computacional' [1, 2], essencial para processar informação e tarefas de forma sistemática e eficiente, não é uma capacidade inata, sendo necessário ser ensinada/aprendida, a quantidade de trabalho que visa perceber de que modo as plataformas podem ser adaptadas a diferentes tipos de utilizadores (com diferentes capacidades de pensamento computacional) é ainda relativamente limitada [3, 4].

Em [5] é feita a primeira abordagem ao desenvolvimento de um método para identificação de problemas de aprendizagem na utilização de plataformas *low code*. O método, apesar de se ter revelado eficaz na previsão de potenciais problemas de interação, é de custo elevado. Sendo um método eminentemente empírico, exige a realização e posterior análise detalhada de múltiplas sessões de teste. Trabalho subsequente identificou a *framework* proposta por Robins et al. [6], em que são caracterizadas as várias dimensões do processo de aprendizagem da programação, como um bom ponto de partida para a análise de plataformas *low code*. Foi ainda identificado um conjunto de questões enfrentadas por não programadores, que devem ser tidas em contas

quando se considera a avaliação de plataformas *low code*.

Partindo do conhecimento já adquirido, nesta proposta pretende-se desenvolver uma abordagem com um cariz mais analítico que permita obter resultados comparáveis aos obtidos em [5]. Para tal, propõe-se a exploração de dados de telemetria capturados pela plataforma para a identificação automática dos momentos de dificuldade por parte dos utilizadores, utilizando informação sobre a frequência e os tipos de interação na plataforma, os movimentos do rato, entre outros.

Pretende-se também verificar da adequabilidade de mapear os conceitos de *Computer Science*, que conhecemos dos processos padrão de desenvolvimento de software, em cada uma das plataformas *low code* a considerar. Desta forma obter-se-á, para cada conceito, a forma como ele é representado e descrito na plataforma, bem assim como o esforço para o comunicar e tornar evidente a perfis que sejam não programadores. Este resultado será muito relevante como medida de escolha da plataforma e análise do esforço de criação de aplicações pelos diferentes perfis.

Por forma a facilitar diferentes tipos de análise, prevê-se o desenvolvimento de diferentes protótipos de plataformas de desenvolvimento (ou a adaptação de plataformas de código aberto existentes) e a comparação dos dados obtidos por vias empírica com os dados obtidos a partir da telemetria recolhida.

5. Referências Bibliográficas

- [1] J.M. Wing (2006) Computational Thinking. *Communications of the ACM* 49:33–35.
- [2] J.J. Lu and G.H. Fletcher (2009) Thinking About Computational Thinking. *ACM SIGCS* [3] B. A. Myers, A. J. Ko, T. D. LaToza and Y. Yoon (2016) Programmers Are Users Too: Human-Centered Methods for Improving Programming Tools. *Computer*, 49(7):44-52, July.
- [4] A. Strawhacker and M.U. Bers (2019) What they learn when they learn coding: investigating cognitive domains and computer programming knowledge in young children. *Educational Technology Research and Development*, 67:541–575.
E Bulletin 41:260–264.
- [5] C. Silva, J. Vieira, J.C. Campos, R. Couto and A.N. Ribeiro (*in press*) Development and validation of a Descriptive Cognitive Model for a Low Code Development Platform. *Human Factors*.
- [6] A. Robins, J. Rountree and N. Rountree (2003) Learning and Teaching Programming: A Review and Discussion, *Computer Science Education*, 13(2):137-172.