

LOW-CODE: O NOVO PARADIGMA DA PROGRAMAÇÃO

Vinícius Oliveira Romano Conceição

Unifametro – Análise e Desenvolvimento de Sistemas

Vinicius_orc@hotmail.com

Eduardo Julião Máximo

Orientador – Professor Msc. Informática Aplicada

Eduardo.maximo@professor.unifametro.edu.br

Área Temática: Engenharia de Software e Computação em Nuvem

Área de Conhecimento: Ciências Tecnológicas

Encontro Científico: XIII Encontro de Iniciação à Pesquisa

Introdução: O setor de desenvolvimento de software está em constante evolução, impulsionado pela necessidade de soluções inovadoras e ágeis. O modelo de programação tradicional, com longos ciclos de codificação manual, está sendo superado por novas metodologias. O Low-Code é uma dessas novas metodologias, que está redefinindo o processo de criação de tecnologia. Ele utiliza plataformas com interfaces visuais, componentes pré-construídos e automação para acelerar a construção de aplicativos. **Objetivo:** O objetivo deste trabalho é apresentar o conceito de Low-Code, demonstrar seus principais benefícios, explorar suas limitações e discutir a sua aplicação e papel complementar no ecossistema de desenvolvimento de software. **Metodologia:** A metodologia utilizada foi a pesquisa bibliográfica, com base em artigos, blogs e documentos de empresas e organizações de pesquisa da área de tecnologia, como IBM, Appian, OutSystems e Pinho, Daniel; Aguiar, Ademar; Amaral, Vasco, para coletar informações sobre o conceito de Low-Code, suas vantagens, desvantagens e áreas de aplicação. **Resultados e Discussão:** O Low-Code se mostra como uma metodologia que reduz a necessidade de escrever milhares de linhas de código, permitindo que profissionais de diversas áreas construam sistemas de forma mais intuitiva. Isso resulta em um ciclo de desenvolvimento significativamente mais rápido, melhorando a produtividade e a capacidade de resposta das empresas. Entre os benefícios tangíveis, destacam-se a aceleração da entrega, a redução de custos e a simplificação da manutenção. No entanto, é crucial estar

ciente das limitações, como a dependência do fornecedor e a menor escalabilidade para sistemas que exigem personalização profunda. A aplicação do Low-Code é vasta, abrangendo desde a criação de aplicativos corporativos até a automação de processos de negócio. **Considerações finais:** O Low-Code não é um substituto para o desenvolvimento tradicional, mas sim um complemento, criando um ambiente onde o Low-Code e o High-Code podem coexistir para atender a diferentes níveis de complexidade e contextos de negócio. Para os profissionais de T.I., o entendimento e a adoção do Low-Code são essenciais para otimizar o tempo e se manterem competitivos em um mercado que exige cada vez mais agilidade e inovação

Palavras-chave: Low-Code; Programação; Metodologia; Desenvolvimento de Software

Referências:

APPIAN. **Low-Code BPM: what it is and why it works.** 22 mar. 2024. Disponível em: <https://appian.com/blog/acp/low-code/low-code-bpm-what-it-is-why-it-works>.

APPIAN. **What is Business Process Management (BPM)?** 21 out. 2015. Disponível em: <https://appian.com/learn/topics/business-process-management/what-is-business-process-management>

APPIAN. **What is low-code automation? How it works.** [S. l.], [s. d.]. Disponível em: <https://appian.com/learn/topics/low-code/low-code-automation>

IBM. **Low-Code vs. No-Code: what's the difference?** 23 maio 2022. Disponível em: <https://appian.com/blog/acp/low-code/low-code-bpm-what-it-is-why-it-works>

IBM. **What is low-code?** [S. l.], [s. d.]. Disponível em: <https://www.ibm.com/think/topics/low-code>

IBM RESEARCH. **Goal-oriented flow assist: supporting low-code data pipelines.** 3 out. 2022. Disponível em: <https://research.ibm.com/blog/GOFA>

OUTSYSTEMS. **Vendor lock-in: understanding risks and how to avoid it.** [S. l.], [s. d.]. Disponível em: <https://www.outsystems.com/application-development/vendor-lock-in-challenges-and-concerns/>

PINHO, Daniel; AGUIAR, Ademar; AMARAL, Vasco. **What about the usability in low-code platforms? A systematic literature review.** Journal of Computer Languages, 2023, p. 101185. DOI: 10.1016/j.cola.2022.101185. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S259011842200082X>