

Resenha de dois artigos sobre Desenvolvimento No-Code e Low-Code

Denys Ferreira Malta

Resumo

Este documento trata-se de uma resenha de dois artigos relacionados ao tema escolhido para meu trabalho de graduação: “**A Comprehensive Overview of No-Code and Low-Code Development Paradigm**” do autor *Joseph Oluwaseyi* e “**LOW CODE / NO CODE DEVELOPMENT PLATFORM COMPARED TO TRADITIONAL DEVELOPMENT FEATURES USES AND FUTURE**” dos autores *Vaibhv Vijay Rajgor* e *Salar Mary*.

O primeiro artigo dos artigos analisados apresenta uma visão abrangente a respeito do desenvolvimento nas modalidades referidas, examinando seus conceitos, vantagens, desafios e aplicações, enquanto o segundo foca o escopo da análise na comparação entre estas novas tecnologias emergentes e os antigos paradigmas de programação já estabelecidos.

Introdução

Desenvolvimento No-Code e Low-Code são definidos como abordagens inovativas que revolucionaram os métodos tradicionais de desenvolvimento de software. Apesar de ser costumeiro que se use os dois termos tentando se referir à mesma coisa, a diferença entre as duas modalidades está no nível de abstração de código que as plataformas podem oferecer. Enquanto uma plataforma No-Code oferece a possibilidade de desenvolver uma aplicação completa apenas usando a interface visual e componentes pré-gerados, uma plataforma Low-Code está no meio termo entre o No-Code e a programação tradicional, agilizando significativamente o processo de desenvolvimento sem abrir mão da customização (visto que este recurso é mais limitado no No-Code, já que os códigos são pré-gerados).

No-Code

A metodologia No-Code define o processo de criar aplicações sem a necessidade de escrever código manualmente. Conta com interfaces visuais e funcionalidade de arratar e soltar os componentes pré-generados da plataforma, facilitando assim o desenvolvimento veloz de softwares.

A trajetória das plataformas No-Code pode ser traçada até o surgimento de linguagens de programação visuais como a Snap!, uma linguagem de programação em blocos, e das próprias IDEs, já que antes das IDEs a programação era feito usando um aplicativo de editor de texto e outro separado para compilar o código escrito. A simplificação oferecida pela IDE ao juntar essas duas funcionalidades em um único app é um princípio também presente na metodologia No-Code.

Hoje, os avanços em áreas como Computação em Nuvem, Inteligência Artificial e UX Design possibilitaram o desenvolvimento de plataformas No-Code mais sofisticadas e com mais recursos,

como automação do processo de desenvolvimento com IA. Um exemplo disso é a plataforma No-Code, que permite o desenvolvimento de aplicativos sem código e com um assistente de IA.

Low-Code

O desenvolvimento low-code refere-se a uma abordagem de desenvolvimento de software que enfatiza ferramentas visuais para acelerar o processo de criação de aplicativos, mas que também oferece a opção da inserção de código manual para que o resultado final possa ser mais customizado. É uma metodologia direcionada a quem já tem conhecimentos sólidos de programação. Um exemplo é a plataforma Tinkercad, que tem componentes prontos para fazer a modelagem de circuitos eletrônicos, e também oferece uma aba onde você pode inserir o seu próprio código nestes circuitos, seja através da codificação manual ou por blocos.

Diferentemente da codificação tradicional, as plataformas low-code permitem que os desenvolvedores criem aplicativos com mínima codificação manual, focando mais na configuração e personalização por meio de uma interface gráfica.

O conceito de desenvolvimento low-code evoluiu de esforços anteriores para simplificar as tarefas de programação e reduzir o tempo necessário para construir software. Ganhou destaque à medida que as organizações buscavam maneiras de atender à crescente demanda por aplicativos enquanto enfrentavam a escassez de desenvolvedores qualificados. As tendências recentes no desenvolvimento low-code incluem a integração de capacidades de IA e aprendizado de máquina para automatizar tarefas repetitivas, recursos aprimorados de colaboração para equipes e suporte para implantação multiplataforma.

Considerações

Ao ler os dois artigos, me elucidei das diferenças entre as abordagens No-Code e Low-Code e seus casos de uso. O No-Code, geralmente, é aplicado para a construção de aplicações internas em empresas, por funcionários que não possuem um expertise em linguagem de programação, já o Low-Code é empregado por funcionários que possuem conhecimentos solidificados de programação para agilizar o processo de desenvolvimento.

Em uma palavra, o que o No-Code/Low-Code representa é automatização. A produtividade e a integração de processos é fortemente beneficiada por essas metodologias, que cria pontes para que áreas distintas estejam envolvidas diretamente em escopos de trabalhos que antigamente seriam designados apenas para a área tecnológica.

Os artigos apontam para a tendência do uso cada vez mais acentuado do No-Code/Low-Code, visto que essas plataformas possibilitam a democratização e aceleração do desenvolvimento de software, respectivamente, otimizando assim o tempo dos trabalhadores, que sob o paradigma tradicional de programação seria empenhado em *hard coding* mesmo para aplicações com requisitos simples. Assim, esses funcionários podem direcionar seus conhecimentos em áreas que ainda não são significativamente atingidas pela automação que o No-Code e Low-Code proporcionam, como o desenvolvimento de softwares complexos e na área de cibersegurança.

Referências

OLUWASEYI, Joseph. [A Comprehensive Overview of No-Code and Low-Code Development Paradigms](#)

VAIBHAV VIJAY, Rajgor. MARY, Sarla. [LOW CODE / NO CODE DEVELOPMENT PLATFORM COMPARED](#)