

INCLUSÃO DIGITAL E DEMOCRATIZAÇÃO DO MACHINE LEARNING: UM ESTUDO BIBLIOMÉTRICO SOBRE FERRAMENTAS LOW/CODE NA FORMAÇÃO DE DESENVOLVEDORES CIDADÃOS

DIGITAL INCLUSION AND THE DEMOCRATIZATION OF MACHINE LEARNING: A BIBLIOMETRIC STUDY ON LOW/NO-CODE TOOLS IN THE TRAINING OF CITIZEN DEVELOPERS

Recebido em: 08/10/2024

Reenviado em: 17/05/2025

Aceito em: 05/06/2025

Publicado em: 23/07/2025

Eduardo Feitosa Bispo¹ 

Universidade Católica de Brasília

Eduardo Amadeu Dutra Moresi² 

Universidade Católica de Brasília

Ana Paula Bernardi da Silva³ 

Universidade Católica de Brasília

Resumo: Nos últimos anos, ferramentas *low/no-code* têm ganhado destaque como resposta à escassez de profissionais qualificados frente à crescente demanda por soluções digitais inovadoras. Este estudo realiza uma análise bibliométrica para investigar como essas soluções, aplicadas ao aprendizado de máquina, vêm sendo integradas às iniciativas de formação de desenvolvedores cidadãos. Essa abordagem permite identificar padrões, tendências e lacunas na literatura, oferecendo uma visão estruturada da evolução temática do campo. Foram analisados 351 documentos da base Scopus, publicados entre 2014 e 2024, com o uso de quatro técnicas bibliométricas: (1) análise de desempenho para mensurar indicadores quantitativos; (2) coocorrência de palavras-chave para mapear conceitos centrais; (3) co-citação para identificar a base teórica do campo; e (4) acoplamento bibliográfico para evidenciar frentes de pesquisa emergentes. Os resultados indicam um crescimento da produção científica, com destaque para 2023, sinalizando o fortalecimento do interesse acadêmico no uso de ferramentas *low/no-code* como estratégia para democratizar o aprendizado de máquina. A análise reforça a importância dessas plataformas na transformação digital e na inclusão de indivíduos não técnicos no desenvolvimento de soluções avançadas.

Palavras-chave: *Low-code, No-code, Machine Learning, Desenvolvedores Cidadãos, Análise Bibliométrica.*

Abstract: In recent years, low/no-code tools have gained prominence as a response to the shortage of qualified professionals amid the growing demand for innovative digital solutions. This study conducts a bibliometric analysis to investigate how these solutions, when applied to machine learning, have been integrated into citizen developer training initiatives. This approach allows for the identification of patterns, emerging trends, and gaps in the literature, providing a structured view of the thematic evolution of the field. A total of 351 documents from the Scopus database, published between 2014 and 2024, were analyzed using four bibliometric techniques: (1) performance analysis to measure quantitative indicators; (2) keyword co-occurrence to map core concepts; (3) co-citation to identify the theoretical foundations of the field; and (4) bibliographic coupling to highlight emerging research fronts. The results indicate a increase in scientific output, especially in 2023, signaling the growing academic interest in low/no-code tools as a strategy to democratize access to machine learning. The analysis

¹ Aluno do Programa de Pós-graduação Stricto Sensu em Governança, Tecnologia e Inovação da Universidade Católica de Brasília. E-mail: eduardo.bispo@a.ucb.br

² Professor do Programa de Pós-graduação em Governança, Tecnologia e Inovação da Universidade Católica de Brasília. E-mail: moresi@p.ucb.br

³ Professora do Programa de Pós-graduação em Governança, Tecnologia e Inovação da Universidade Católica de Brasília. E-mail: anapaula@p.ucb.br

reinforces the importance of these platforms in driving digital transformation and in promoting the inclusion of non-technical individuals in the development of advanced solutions.

Keyword: Low-code, No-code, Machine Learning, Citizen Developers, Bibliometric Analysis.

INTRODUÇÃO

Nos últimos anos, o cenário tecnológico tem sido marcado por um descompasso entre a demanda por soluções digitais inovadoras e a escassez de profissionais qualificados (IBM, 2022). Nesse contexto, as ferramentas *low/no-code* emergem como alternativas promissoras, permitindo que indivíduos sem formação técnica desenvolvam soluções computacionais com maior autonomia e agilidade.

Essas plataformas transformam a codificação tradicional em interfaces visuais baseadas em lógica de arrastar e soltar, facilitando o uso de aplicações por profissionais de diferentes áreas, sem exigir domínio avançado de programação, de acordo com Sundberg e Holmström, 2023.

O termo *low-code* ganhou notoriedade a partir do relatório de Richardson e Rymer (2014), que destacou seu potencial para acelerar a transformação digital ao simplificar o desenvolvimento de sistemas. Posteriormente, Atkins (2021) associou essas tecnologias ao surgimento dos chamados desenvolvedores cidadão — indivíduos não técnicos capazes de criar soluções digitais por meio dessas plataformas.

A relevância das ferramentas *low/no-code* para a inovação organizacional e a eficiência operacional é amplamente reconhecida na literatura (Picek, 2023; El Kamouchi et al., 2023; Razak et al., 2024). Estudos recentes também exploram seu papel na democratização do acesso à inteligência artificial (Curty et al., 2023; Rokis; Kirikova, 2022; Favre et al., 2023) e na integração com tecnologias emergentes como aprendizado de máquina e automação de processos (Hurlburt, 2021; Sundberg; Holmström, 2024; Liu; Downing, 2024).

Diante desse panorama, este artigo investiga como as soluções *low/no-code* voltadas ao aprendizado de máquina têm sido integradas às iniciativas de formação de desenvolvedores cidadãos. Mais especificamente, busca responder à questão: de que maneira as soluções *low-code/no-code* para aprendizado de máquina estão sendo integradas às iniciativas de formação de desenvolvedores cidadãos?

Ao mapear e analisar as interconexões entre *low-code*, aprendizado de máquina (*machine learning*) e desenvolvedores cidadãos (*citizen developers*), este estudo fornece subsídios para que organizações adotem práticas mais inclusivas de desenvolvimento tecnológico. A análise permite identificar boas práticas, barreiras e facilitadores associados à

formação de desenvolvedores cidadãos por meio da democratização de ferramentas de aprendizado de máquina.

Este artigo apresenta uma revisão bibliográfica com base na análise bibliométrica/cientométrica — abordagem quantitativa que identifica padrões, tendências e lacunas em grandes volumes de publicações científicas, permitindo mapear a evolução temática de um campo e avaliar a contribuição de seus principais agentes (Donthu *et al.*, 2021).

A escolha dessa metodologia se justifica pelo objetivo central da pesquisa, que é compreender como as ferramentas *low/no-code* de aprendizado de máquina vêm sendo abordadas na literatura científica no contexto da formação de desenvolvedores cidadãos.

PROCEDIMENTOS METODOLÓGICOS

A metodologia deste trabalho está organizada em três itens principais: a coleta de dados, o refinamento da pesquisa e a análise bibliométrica.

COLETA E ANÁLISE DE DADOS

Para a coleta de dados, foi conduzida uma pesquisa inicial na base Scopus utilizando a expressão de busca: “*low-code* OR "low code*" OR no-code* OR "no code*"*”. Os resultados foram delimitados aos tipos de documentos publicados em periódicos, conferências ou revisões, e na língua inglesa. Esta busca resultou o arquivo bruto com 1549, com um pico de publicações em 2023 (320 documentos).

Com a coleta de dados finalizada, foi iniciado o processo de refinamento.

REFINAMENTO DA PESQUISA

Duas ferramentas foram utilizadas na etapa de refinamento dos dados.

O VOSviewer — software voltado ao mapeamento e visualização de redes bibliométricas — foi utilizado para identificar coocorrência entre palavras-chave, a partir dos metadados exportados em formato CSV, com base na unidade "Author Keywords" (Van Eck; Waltman, 2010). A técnica de coocorrência examina a frequência com que determinados termos aparecem juntos em diferentes publicações, permitindo identificar padrões de associação temática e revelar os principais conceitos em circulação dentro do campo de estudo (Donthu *et al.* (2021). A ferramenta está disponível em: <https://www.vosviewer.com>.

Adotou-se um limiar mínimo de duas ocorrências por termo, resultando em 286 palavras-chave. Utilizou-se também um arquivo tesouro, que padroniza variações

terminológicas para garantir consistência na análise. Tesouros são amplamente usados em estudos informétricos para controlar o vocabulário e aumentar a precisão dos resultados (Soergel, 1995).

A partir daí, o arquivo. gml foi exportado para uso no Gephi — software livre voltado à análise e visualização de redes complexas (Bastian; Heymann; Jacomy, 2009). A ferramenta está disponível em: <https://gephi.org>.

Foi selecionado o tipo de grafo não dirigido, uma vez que essa configuração permite considerar as conexões entre os nós como relações bidirecionais. Ou seja, a ligação entre dois termos é analisada independentemente do sentido da aresta, o que é especialmente útil para representar relações de coocorrência, onde a reciprocidade entre os termos é relevante para a interpretação dos dados.

Na aba arestas, os dados foram inicialmente ordenados pela coluna "Origem" e as linhas contendo os termos "*low-code*" e "*no-code*" foram selecionadas e armazenadas em uma planilha eletrônica. O processo foi repetido para a coluna "Destino". A Tabela 1 apresenta uma amostra dos dados resultantes, organizados de acordo com o peso das conexões.

Tabela 1 – Amostra de termos para refinamento da busca.

ORIGEM	DESTINO	PESO
1948 - low-code	2288 - no-code	16.0
1948 - low-code	2160 - model-driven engineering	8.0
1948 - low-code	3144 - software development	6.0
1948 - low-code	2085 - microservices	5.0
1948 - low-code	1989 - machine learning	4.0
1948 - low-code	2155 - model-driven	4.0
1948 - low-code	2443 - outsystems	4.0
2288 - no-code	3144 - software development	4.0
1948 - low-code	3089 - smart contracts	3.0
2288 - no-code	3510 - user experience	3.0

Elaboração: Do autor (2024). **Fonte:** Gephi.

A tabela resultante foi utilizada para escolher os termos para refinar a busca original na Scopus, a seleção foi pautada em 3 premissas: Ferramentas *low/no-code*, técnicas de aprendizado de máquina e relação com desenvolvedor cidadão.

Em sequência, os termos selecionados foram sendo incluídos no campo de refinamento na página Scopus até atingir um nível de saturação, conforme apresentado na Tabela 2.

Tabela 2 – Termos utilizados para refinar a busca na Scopus.

PALAVRAS-CHAVE DO REFINAMENTO	QUANTIDADE DE ARQUIVOS
Busca original	1517
"low/no code"	13
"low/no code" OR "machine learning"	273
"low/no code" OR "machine learning" OR "deep learning"	305
"low/no code" OR "machine learning" OR "deep learning" OR "automated machine learning"	305
"low/no code" OR "machine learning" OR "deep learning" OR "automated machine learning" OR "end-user development"	331
"low/no code" OR "machine learning" OR "deep learning" OR "automated machine learning" OR "end-user development" OR "end-user programming"	335
"low/no code" OR "machine learning" OR "deep learning" OR "automated machine learning" OR "end-user development" OR "end-user programming" OR "citizen development"	348
"low/no code" OR "machine learning" OR "deep learning" OR "automated machine learning" OR "end-user development" OR "end-user programming" OR "citizen development" OR "user-centered design"	353
"low/no code" OR "machine learning" OR "deep learning" OR "automated machine learning" OR "end-user development" OR "end-user programming" OR "citizen development" OR "user-centered design" OR "conversational user interfaces"	355
"low/no code" OR "machine learning" OR "deep learning" OR "automated machine learning" OR "end-user development" OR "end-user programming" OR "citizen development" OR "user-centered design" OR "conversational user interfaces" OR "data science democratization"	355

Elaboração: Do autor (2024). **Fonte:** Gephi.

Após o refinamento, a busca resultou em 355 documentos publicados entre 2005 e 2024.

Como já abordado no presente artigo, o termo *low-code* surgiu no contexto da abstração tecnológica em 2014, desta forma, foi aplicado o filtro para excluir as publicações anteriores ao ano citado. O resultado foi de 351 documentos.

Por conseguinte, foi realizada a exportação dos metadados para um arquivo .csv para ser realizada a análise bibliométrica.

ANÁLISE BIBLIOMÉTRICA

As técnicas escolhidas para esta análise foram: análise de desempenho, para oferecer uma visão quantitativa da produção científica; coocorrência de palavras-chave, para o mapeamento conceitual; co-citação, para identificar a estrutura intelectual do campo; e acoplamento bibliográfico, para explorar a proximidade temática entre os documentos analisados e frentes de pesquisa.

ANÁLISE DE DESEMPENHO

Trata-se de uma etapa da bibliometria voltada à mensuração da produção científica de um campo de estudo, por meio de indicadores quantitativos. Para isso, foi utilizado o pacote Bibliometrix, da linguagem R, desenvolvido especificamente para análises bibliométricas. A ferramenta permite a extração de métricas como, por exemplo a produtividade por autor, periódico e país (Aria; Cuccurullo, 2017). A ferramenta está disponível em: <https://www.bibliometrix.org/home/>.

A Tabela 3 apresenta uma amostra das informações gerais do *corpus*, destaca-se a taxa de crescimento anual das publicações em 53,02%, esse fato destaca a ascensão e a importância da temática aqui pesquisada. Esse percentual foi calculado com base na diferença entre o número de publicações no início e no final do período (2015 a 2024), considerando a média de crescimento ano a ano, conforme fórmula de crescimento composto utilizada pelo próprio *software*.

Tabela 3 – Amostra de desempenho da Análise Bibliométrica.

DESCRIÇÃO	RESULTADOS
INFORMAÇÕES GERAIS SOBRE OS DADOS	
Intervalo de tempo	2015–2024
Fontes (Periódicos, Livros etc.)	256
Documentos	351
Taxa de crescimento anual (%)	53,02
Idade média dos documentos	1,72
Média de citações por documento	5,031
Referências	12.469

Elaboração: Do autor (2024). **Fonte:** Bibliometrix.

Quando o conjunto de metadados é analisado no Bibliometrix, a métrica *Most Local Cited Documents* permite identificar os artigos mais citados dentro do próprio *corpus* analisado, ou seja, os que exercem maior influência entre os documentos da amostra. Com base nesse critério, foram selecionados os 10 artigos com maior número de citações locais.

Esses documentos foram então organizados em três grupos temáticos de acordo com seus respectivos focos de investigação: (1) desafios e práticas no desenvolvimento *low-code/no-code*; (2) aspectos conceituais e metodológicos relacionados à engenharia *low-code/no-code*; e (3) estratégias de adoção e fatores organizacionais, conforme apresentado na Tabela 4. Essa organização permite compreender as abordagens predominantes na literatura mais influente da área, reforçando os eixos que sustentam a formação de desenvolvedores cidadãos por meio de ferramentas *low/no-code* aplicadas ao aprendizado de máquina.

Tabela 4 – Grupos dos 10 documentos mais citados no corpus.

TÍTULO	AUTOR(ES)	ANO	TOTAL CITAÇÕES
Grupo 1 - Desafios e Práticas em Desenvolvimento Low-code			
Low-Code Is Often High-Code, So We Must Design Low-Code Platforms to Enable Proper Software Engineering	Lethbridge T. C.	2021	10
An Empirical Study of Developer Discussions on Low-Code Software Development Challenges	Al Alamin M. A.; Malakar S.; Uddin G.; Afroz S.; Haider T. B.; Iqbal A.	2021	9
Challenges of Low-Code/No-Code Software Development: A Literature Review	Rokis K.; Kirkova M.	2021	5
Proposing a Framework for Impact Analysis for Low-Code Development Platforms	Overeem M.; Jansen S.	2021	5
Grupo 2 - Estudos Exploratórios e de Modelagem que Abordam Low-code			
In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms	Bock A. C.; Frank U.	2021	5
Modelling in Low-Code Development: A Multi-Vocal Systematic Review	Bucaioti A.; Cicchetti A.; Ciccozzi F.	2022	5
Low-Code Programming Models	Hirzel M.	2023	4
Grupo 3 - Adoção e Usabilidade de Low Code			
Empower the Workforce, Empower the Company? Citizen Development Adoption	Hoogsteen D.; Borgman H.	2022	6
Drivers and Inhibitors of Low-Code Development Platform Adoption	Kass S.; Strahringer S.; Westner M.	2022	3
Development and Validation of a Descriptive Cognitive Model for Predicting Usability Issues in a Low-Code Platform	Silva C.; Vieira J.; Campos J. C.; Couto R.; Ribeiro A. N.	2023	3

Elaboração: Do autor, 2024. **Fonte:** Bibliometrix.

Sobre os trabalhos que tratam os desafios e práticas em desenvolvimento *low-code*, Lethbridge (2021) discute a necessidade de projetar plataformas essas plataformas para evitar a acumulação de dívida técnica — o autor explica que o termo se refere ao acúmulo de problemas técnicos ou decisões de curto prazo que comprometem a qualidade e a manutenção do *software* no futuro. Al Alamin *et al.* (2021) realizaram um estudo empírico sobre discussões de desenvolvedores em fóruns, identificando desafios significativos relacionados à personalização e adoção de plataformas *low-code*. Rokis e Kirikova (2022) revisaram a literatura existente sobre desenvolvimento *low-code/no-code*, identificando os principais desafios e propondo direções para melhorias futuras. Overeem e Jansen (2021) propuseram uma estrutura de análise de impacto para plataformas *low-code*, auxiliando desenvolvedores e engenheiros a entenderem as implicações das mudanças no sistema.

Nos estudos exploratórios e de modelagem que abordam *low-code*, Bock e Frank (2021) conduziram uma análise sobre sete plataformas, avaliando suas características e destacando a

combinação inovadora de funcionalidades. Bucaioni, Cicchetti e Ciccozzi (2022) realizaram uma revisão sistemática multi-vocal, explorando sua relação com a engenharia dirigida por modelos e oferecendo uma visão estruturada sobre o estado atual da tecnologia. Hirzel (2023) destacou o potencial para capacitar mais pessoas a automatizarem tarefas, facilitando a criação de programas de computador por usuários não técnicos

Quando se trata de adoção e usabilidade, Kass, Strahringer e Westner (2022) identificaram fatores que impulsionam e inibem a adoção de plataformas *low-code*, destacando lacunas na literatura e sugerindo direções para pesquisas futuras. Silva *et al.* (2020) desenvolveram e validaram um modelo cognitivo descritivo para prever problemas de usabilidade em plataformas *low-code*, identificando diferentes padrões de problemas enfrentados por especialistas e novatos. O estudo de Hoogsteen e Borgman (2022) explora a adoção de ferramentas de *low-code* na formação de desenvolvedores cidadãos, a pesquisa foi balizada na entrevista com profissionais de 6 grandes organizações.

Esses agrupamentos permitiram uma visão sobre as possíveis estruturas que as pesquisas nessa área estão seguindo. Para o presente trabalho, o GRUPO 3 – Adoção e Usabilidade têm grande relevância.

Contudo, cabe destacar que nesse grupo os artigos não abordam diretamente como as ferramentas *low-code* de aprendizado de máquina estão sendo utilizadas para formação de desenvolvedores cidadãos. Desta forma, parece existir um campo pouco explorado nesse sentido.

ANÁLISE DA ESTRUTURA CONCEITUAL

Também utilizando o Bibliometrix foi realizada a análise da estrutura conceitual, que tem como objetivo identificar os principais temas e conceitos que organizam o campo de estudo. Essa técnica foi empregada por meio da coocorrência de palavras-chave, a qual examina a frequência com que determinados termos aparecem juntos em diferentes publicações, permitindo identificar padrões de associação temática e revelar os principais conceitos em circulação dentro do campo de estudo.

Ao analisar a frequência e a associação entre palavras-chave nos documentos, é possível revelar a dinâmica conceitual do campo, suas áreas de convergência e os focos de investigação predominantes na pesquisa.

A primeira análise foi sobre os tópicos de tendências (*trend topics*) que fornecem uma visão detalhada dos principais termos e conceitos discutidos na literatura relacionada e permite compreender quando esses temas ganharam destaque na literatura acadêmica.

Ao configurar essa opção com a frequência mínima de palavras em 5, número de palavras por ano em 10 e organizada pela maior média anual das publicações, foi possível extrair a Tabela 5 com as 10 palavras de maior destaque.

Tabela 5 – Tópicos de tendência.

ITEM	FREQUÊNCIA	ANO INÍCIO	MÉDIA ANUAL	ANO RECENTE
machine learning	52	2022	2023	2023
low-code	49	2022	2023	2023
artificial intelligence	42	2022	2023	2023
no-code	29	2022	2023	2023
low-code development	19	2022	2023	2023
citizen developer	13	2022	2023	2023
end-user development	9	2022	2023	2023
cloud	8	2023	2023	2023
no-code development	6	2022	2023	2023
deep learning	25	2021	2022	2023

Elaboração: Do autor (2024). **Fonte:** Bibliometrix, 2024.

Dentre os termos é possível verificar que *low-code*, *machine learning* e *no-code* aparecem com grande destaque, o que corrobora o entendimento que esses assuntos estão estabelecidos como campos atuais de pesquisas acadêmicas.

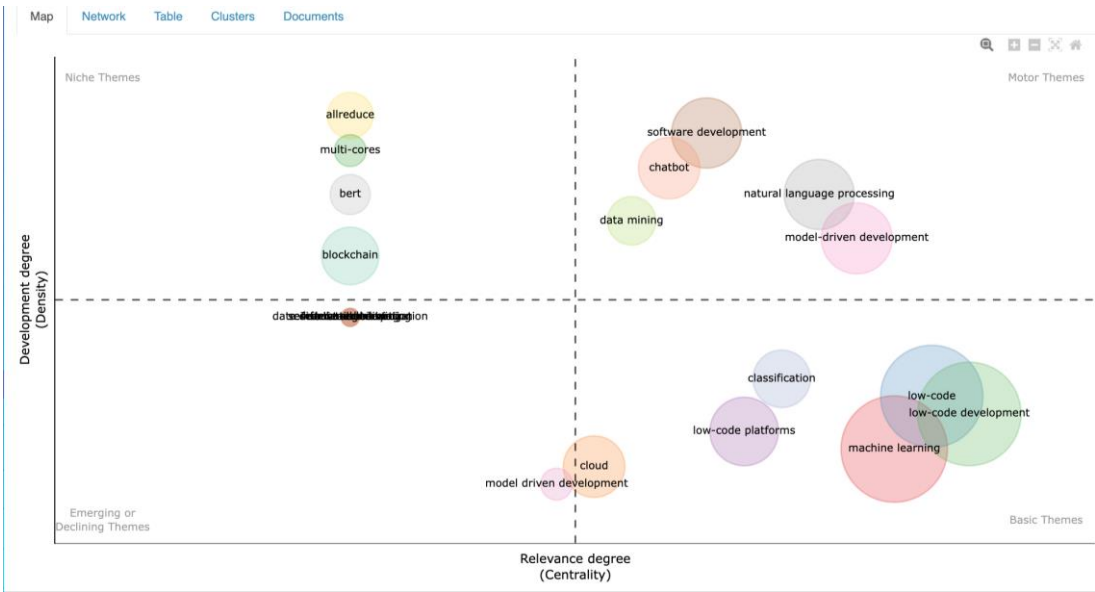
Na lista aparece o termo *Citizen developer*, mesmo com uma frequência menor comparada aos outros tópicos, sua presença reflete o foco emergente na capacitação de indivíduos não técnicos para desenvolver e utilizar aplicações que abstraíam a codificação, uma tendência que indica um campo de pesquisa promissor.

O Mapa Temático é uma ferramenta de análise bibliométrica que representa visualmente a importância e a maturidade de diferentes tópicos de pesquisa com base em dois eixos: densidade (grau de desenvolvimento) e centralidade (grau de relevância). Ao gerar esse mapa (Imagem 4) no Bibliometrix foi possível identificar como está o campo de pesquisa que relaciona *low-code/no-code*, *machine learning* e *citizen developer*. Os parâmetros utilizados para gerar a visualização foram: Número de palavras: 1033; Frequência mínima de documentos por cluster: 1; Número de rótulos: 1; Tamanho do rótulo: 0,1; Repulsão: 0; Algoritmo de agrupamento: Louvain.

Ao analisar os resultados da Imagem 1 é possível identificar, no quadrante direito inferior, os assuntos que são centrais para pesquisas atuais, mas ainda necessitam de mais estudo e aprofundamento. Nesse espaço estão os termos *Low-code*, *Low-code development* e *Machine Learning*, e as interseções.

Ao clicar na representação de cada grupo de interesse é possível verificar os subtemas pesquisados, a Tabela 6 ilustra a divisão. Vale destacar que devido a aleatoriedade do algoritmo, podem existir pequenas variações nesses grupos caso o processo seja repetido.

Imagem 1 –Mapa Temático.



Fonte: Bibliometrix, 2024.

Tabela 6 – Subtemas de interesse no Mapa Temático.

Machine Learning	Low-code	Low-code Development
machine learning 52 artificial intelligence 41 deep learning 25 automated machine learning 7 automation 4 industry 4.0 4 no-code platforms 3 smart manufacturing 3 code smell 2 convolutional neural network 2	low-code 49 no-code 29 iot 16 end-user programming 6 information system 4 data analysis 3 software 3 spreadsheets 3 adoption 2 api 2	low-code development 19 low-code development platforms 18 model-driven engineering 13 citizen developer 11 end-user development 9 domain-specific language 8 no-code development 6 requirements 6 software engineering 6 case study 4

Elaboração: Do autor (2024). Fonte: Bibliometrix, 2024.

No trabalho publicado por Hoogsteen e Borgman (2022), eles destacam a evolução dos conceitos até chegar ao desenvolvedor cidadão. Na pesquisa, é explicado que esse desenvolvedor cidadão é uma extensão do termo usuário final (*end-user*). A principal diferença apontada é que o desenvolvimento por usuário final é focado em uma única pessoa, enquanto o

desenvolvimento cidadão se concentra em capacitar grupos de indivíduos que não são da área de tecnologia, para desenvolver aplicativos por meio de ferramentas *low-code*.

Com esse panorama, é possível entender que os conceitos *end-user* e *citizen developer* fazem parte do mesmo contexto, o que traz a ligação entre os grupos *Low-code* e *Low-code Development*.

Sunderberg e Holmstrom (2023) explicam o potencial das ferramentas *low-code* de aprendizado na democratização da inteligência artificial. Eles descrevem que essas ferramentas permitem transformar usuários sem conhecimento em codificação em desenvolvedores cidadãos. Já Raghavendran e Elragal (2023) publicaram um estudo sobre como as ferramentas de *low-code* de aprendizado de máquina podem acelerar a digitalização das instituições. Eles fazem referência ao termo aprendizado de máquina automatizado (*automated machine learning*); e reforçam que essas ferramentas são acessíveis até para usuários sem conhecimentos técnicos, mesmo que não citado diretamente, o entendimento é de que essas ferramentas são utilizadas por desenvolvedores cidadãos.

Ademais, no estudo de Ashraf *et al.* (2024), eles afirmam que é crucial empregar uma abordagem centrada no ser humano, treinando os usuários finais sobre como interagir adequadamente com o sistema baseado em *deep learning* para maximizar benefícios para a corporação.

O termo *automated machine learning* se vincula ao desenvolvedor cidadão por meio dos termos requisitos, *low-code* e *no-code*. Desta forma, é possível afirmar que um dos elos entre os grupos analisados são pessoas que podem ser consideradas desenvolvedores cidadãos, visto que uma característica intrínseca dos estudos analisados é a abstração da codificação. Essa análise conceitual traz a reflexão de que na literatura não há um conceito único para definir os usuários não especialistas, que utilizam soluções *low-code* para realizar trabalhos que antes necessitavam de conhecimentos técnicos em codificação.

Outro ponto de ponderação é que a evolução do uso de ferramentas *low-code* pode estar convergindo conceitos para o desenvolvedor cidadão. Com isso, é possível verificar que o termo desenvolvedor cidadão aparece com destaque na presente análise bibliométrica, porém, devido as características convergentes dos conceitos *end-user programming*, *end-user development*, *citizen developer* e *automated machine learning*, se aglutinar essas visões de abstração de codificação, o impacto desse assunto tende a ser mais relevante.

Por fim, os resultados apresentados na análise conceitual realizada por meio da Nuvem de palavras, *Trends topics* e Mapa temático, demonstram centralidade do termo *low-code* e *Machine Learning* e a caracterização de *Citizen developer* como assunto emergente.

ANÁLISE DA ESTRUTURA INTELECTUAL

Após o mapeamento dos principais conceitos e suas inter-relações, a análise seguiu para a identificação da estrutura intelectual do campo, com o objetivo de reconhecer os autores e obras que fundamentam teoricamente a produção científica sobre plataformas *low/no-code* e a formação de desenvolvedores cidadãos.

Para isso, utilizou-se a técnica de co-citação, que considera as referências citadas pelos artigos da base, revelando autores frequentemente mencionados em conjunto. Quanto maior a frequência de co-citação entre dois autores, maior a probabilidade de afinidade temática ou alinhamento teórico entre eles. As redes foram elaboradas com o auxílio dos *softwares* VOSviewer e Gephi.

Na primeira ferramenta foi realizada a importação, a configuração aplicada estabeleceu um limiar mínimo de três citações por referência, resultando em 62 documentos que atendem a este critério, de um total de 12.205 referências citadas. Foram identificados 7 grupos de conexões intelectuais.

A partir daí um arquivo. gml foi exportado para análise dentro do Gephi, então foram selecionados os 3 artigos com maior Centralidade do Autovetor (CAV) por grupo (Tabela 7) – essa métrica identifica os nós mais influentes da rede com base na força de suas conexões e na relevância dos nós aos quais estão ligados. Esse indicador permite destacar as referências que têm influência central na estrutura temática analisada.

Tabela 7 – Grupos de influências e conexões entre autores

Referência	Grupo	CAV
waszkowski r., low-code platform for automating business processes in manufacturing, ifac-papersonline, 52, 10, pp. 376-381, (2019)	1	1,00
sahay a., indamutsa a., di ruscio d., pierantonio a., supporting the understanding and comparison of low-code development platforms, 2020 46th euromicro conference on software engineering and advanced applications (seaa), pp. 171-178, (2020)	1	0,93
daniel g., cabot j., deruelle l., derras m., xatkit: a multimodal low-code chatbot development framework, iee access, 8, pp. 15332-15346, (2020)	1	0,64
sanchis r., garcia-perales o., fraile f., poler r., low-code as enabler of digital transformation in manufacturing industry, appl. sci., 10, 1, pp. 1-17, (2020)	2	0,88
bock a.c., frank u., low-code platform, bus. inf. syst. eng, 63, pp. 733-740, (2021)	2	0,72

varajao j., software development in disruptive times, <i>commun. acm</i> , 64, 10, pp. 32-35, (2021)	2	0,52
lethbridge t. c., low-code is often high-code, so we must design low-code platforms to enable proper software engineering, <i>international symposium on leveraging applications of formal methods</i> , pp. 202-212, (2021)	3	0,56
barricelli b.r., cassano f., fogli d., piccinno a., end-user development, end-user programming and end-user software engineering: a systematic mapping study, <i>j. syst. softw.</i> , 149, pp. 101-137, (2019)	3	0,43
webster j., watson r. t., analyzing the past to prepare for the future: writing a literature review, <i>mis quarterly</i> , 26, 2, pp. xiii-xxiii, (2002)	3	0,39
richardson c., rymer j.r., mines c., cullen a., whittaker d., new development platforms emerge for customer-facing applications, <i>forrester</i> , (2014)	4	0,80
woo m., the rise of no/low code software development—no experience needed?, <i>engineering</i> , 6, 9, pp. 960-961, (2020)	4	0,77
di ruscio d., kolovos d., de lara j., pierantonio a., tisi m., wimmer m., low-code development and model-driven engineering: two sides of the same coin?, <i>softw. syst. model.</i> , (2022)	4	0,67
richardson c., rymer j.r., vendor landscape: the fractured, fertile terrain of low-code application platforms, (2016)	5	0,53
lieberman h., paterno f., klann m., wulf v., end-user development: an emerging paradigm, <i>end user development</i> , pp. 1-8, (2006)	5	0,48
woo m., the rise of no/low code software development-no experience needed?, <i>engineering</i> , 6, 9, pp. 960-961, (2020)	5	0,33
hirzel m., low-code programming models, <i>communications of the acm</i> , 66, 10, pp. 76-85, (2023)	6	0,34
gpt-4 technical report, (2023)	6	0,11
raffel colin, shazeer noam, roberts adam, lee katherine, narang sharan, matena michael, zhou yanqi, li wei, liu peter j., exploring the limits of transfer learning with a unified text-to-text transformer, <i>journal of machine learning research</i> , 21, 140, pp. 1-67, (2020)	6	0,01
goodfellow i., bengio y., courville a., deep learning, (2016)	7	0,10
breiman l., random forests, <i>machine learning</i> , 45, pp. 5-32, (2001)	7	0,03
mikolov t., chen k., corrado g., dean j., efficient estimation of word representations in vector space, (2013)	7	0,01

Elaboração: Do autor, 2024. **Fonte:** GEPHI, 2024.

Devido a interseção de assuntos, os estudos do grupo 1 foram definidos como Plataformas *Low-Code* para Automação de Processos e Desenvolvimento de Aplicações. Nesse grupo, Waszkowski (2019) destacou o uso de plataformas *low-code* para automação de processos de negócios na manufatura, utilizando a plataforma Aurea BPM (Disponível em: <https://www.tecna.pl/>). Sahay *et al.* (2020) realizaram um estudo técnico comparativo de diferentes plataformas de desenvolvimento *low-code* identificando e classificando suas funcionalidades para facilitar a escolha da plataforma mais adequada conforme as necessidades dos usuários. Por fim, Daniel *et al.* (2020) introduziram o Xatkit, um *framework* multimodal *low-code* para desenvolvimento de *chatbots* (Disponível em: <https://xatkit.com/>).

O grupo 2 foi nomeado de Transformação Digital e Indústria 4.0. Sanchis, Perales e Fraile (2019) investigaram como plataformas *low-code* promovem a transformação digital na indústria de manufatura, utilizando a plataforma vf-OS como exemplo (Disponível em: <https://innovation-radar.ec.europa.eu/innovation/36748>). Bock e Frank (2021) analisaram as plataformas *low-code* emergentes, abordando suas características distintivas e comparando-as com tecnologias existentes. Varajão (2021) discutiu a criação de soluções de software em tempos disruptivos utilizando tecnologia *low-code* extrema.

De acordo com o conteúdo dos estudos do grupo 3, eles foram categorizados como Desenvolvimento por Usuários Finais e Engenharia de Software. Lethbridge (2021) destaca a importância de projetar plataformas *low-code* que minimizem a dívida técnica. Para evitar esse cenário, o autor recomenda a adoção de boas práticas de engenharia de software, como controle de versão e testes automatizados. Nesse contexto, Barricelli *et al.* (2019) realizaram um mapeamento sistemático sobre o desenvolvimento por usuários finais, abordando práticas e desafios relacionados à programação e engenharia de software nesse perfil de desenvolvedor. Webster e Watson (2002) destacaram a importância da revisão de literatura como fundamento para o avanço do conhecimento acadêmico, facilitando o desenvolvimento de teorias e identificando áreas que necessitam de mais pesquisa.

Os estudos do grupo 4 foram chamados de Plataformas *Low-Code* para Aplicações Voltadas ao Cliente. Richardson e Rymer (2014) analisaram o surgimento de plataformas de desenvolvimento *low-code* de aplicativos por não técnicos. Woo (2020) discutiu o rápido desenvolvimento e implementação de plataformas *no-code* e *low-code*, que permitem a criação de soluções com mínima ou nenhuma codificação. Di Ruscio *et al.* (2022) compararam as abordagens de desenvolvimento *low-code* com a engenharia dirigida por modelos (MDE), discutindo suas diferenças e semelhanças.

O grupo 5 é sobre o Fornecimento e adoção de ferramentas *Low-code*. Richardson e Rymer (2016) examinaram a paisagem fragmentada e fértil dos fornecedores de plataformas de desenvolvimento *low-code*. Lieberman *et al.* (2006) já discutiam o paradigma emergente do desenvolvimento por usuários finais, abordando a necessidade de sistemas interativos que não apenas sejam fáceis de usar, mas também fáceis de desenvolver por usuários finais. Woo (2020) analisou a ascensão do desenvolvimento *no-code* e *low-code*, enfatizando a importância dessas plataformas na criação rápida de soluções, especialmente durante a pandemia de COVID-19.

Grupo 6 é sobre Modelos de Programação *Low-Code* e Transferência de Conhecimento. Hirzel (2023) explorou os modelos de programação *low-code*, destacando seu potencial para

capacitar mais pessoas a automatizar tarefas e criar programas de computador. OpenAI et al. (2023) relataram o desenvolvimento do GPT-4, um modelo multimodal de grande escala capaz de aceitar entradas de texto e imagem e produzir saídas de texto. Raffel *et al.* (2019) investigaram os limites da aprendizagem por transferência com um transformador de texto para texto unificado, destacando a eficácia dessa técnica em várias tarefas de processamento de linguagem natural (NLP).

O agrupamento 7 trata de Aprendizagem Profunda e Modelos de *Machine Learning*. Goodfellow, Bengio e Courville (2016) introduziram uma ampla gama de tópicos em aprendizagem profunda, cobrindo o background matemático e conceitual, técnicas de *deep learning* usadas na indústria e perspectivas de pesquisa. Breiman (2001) apresentou o algoritmo *Random Forests*, uma técnica de *machine learning* que é crucial para a classificação e predição em grandes conjuntos de dados. Mikolov *et al.* (2013) propuseram arquiteturas de modelos para computação de representações vetoriais contínuas de palavras, melhorando significativamente a precisão na tarefa de similaridade de palavras.

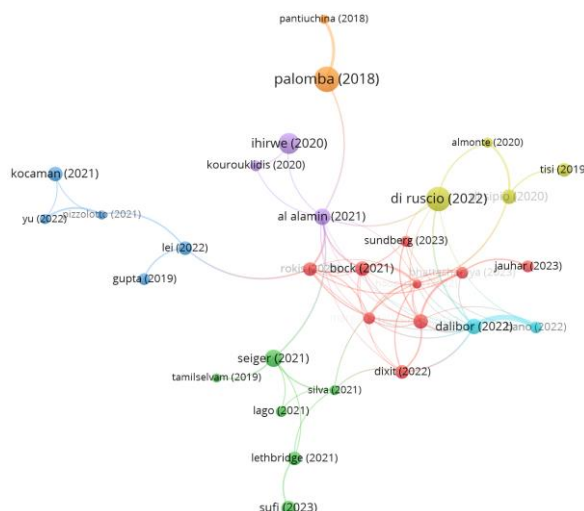
A análise da estrutura intelectual revela um conjunto de referências que serve de alicerce teórico para as pesquisas em *low-code*, aprendizagem de máquina e desenvolvedores cidadãos. Embora ainda tratadas de forma dispersa, essas bases podem ser integradas para sustentar investigações como a deste estudo, voltadas à formação de desenvolvedores cidadãos com foco em aprendizado de máquina e inclusão digital.

ACOPLAMENTO BIBLIOGRÁFICO

O acoplamento bibliográfico é uma técnica que identifica relações temáticas entre documentos que compartilham as mesmas referências, permitindo evidenciar frentes de pesquisa em formação. Diferentemente da co-citação, que revela a base teórica ao identificar artigos frequentemente citados em conjunto, o acoplamento destaca estudos atuais.

Com o apoio do VOSviewer, a técnica de acoplamento bibliográfico foi aplicada para identificar grupos de artigos que compartilham uma base de referências comum, o que pode sugerir afinidades temáticas e aproximações conceituais. Considerando apenas documentos com, no mínimo, dez citações, foram identificados 55 registros, dos quais 31 apresentaram conexões relevantes. A Imagem 2 ilustra a rede formada, na qual foram identificados 7 grupos distintos, representando possíveis frentes de pesquisa no campo analisado.

Imagem 2 – Rede de Acoplamento Bibliográfico



Fonte: VOSviewer (2024).

Após a análise inicial de acoplamento bibliográfico realizada no VOSviewer, o arquivo. gml foi exportado e a Imagem 3 foi gerada a partir dos dados extraídos do Gephi. Ela apresenta os artigos agrupados com base em suas correlações e temas comuns, baseado na centralidade do Autovetor. Cada agrupamento reflete similaridades nos tópicos abordados, metodologias utilizadas e objetivos de pesquisa.

Após a análise do título, palavras-chave e resumo de cada artigo, buscou-se atribuir para cada grupo uma temática central, com o objetivo de identificar as possíveis frentes de pesquisas no *corpus*.

O Grupo 1 aborda os fundamentos e desafios do desenvolvimento *low-code*, destacando como essas plataformas, baseadas em abstração e automação, combinam tecnologias existentes para oferecer abordagens eficientes e acessíveis. Também são discutidas limitações enfrentadas por desenvolvedores em ambientes pouco customizáveis.

O Grupo 2 trata da aplicação de plataformas *low-code* em contextos como Internet das Coisas (IoT) e aprendizado profundo. Os estudos evidenciam como essas tecnologias viabilizam soluções eficazes para análise de desempenho e operação de sistemas complexos com mínima necessidade de codificação tradicional.

Imagem 3 – Grupos de interconexões entre documentos.

Label	cluster	Eigenvector centrality	Title
bucaloni (2022)	1	1	1 Modeling in low-code development: a multi-vocal systematic review
hoogsteen (2022)	1	0,9001	Empower the Workforce, Empower the Company? Citizen Development Adoption
bock (2021)	1	0,8548	In Search of the Essence of Low-Code: An Exploratory Study of Seven Development Platforms
margaria (2021)	1	0,8548	The Interoperability Challenge: Building a Model-Driven Digital Thread Platform for CPS
rokis (2022)	1	0,841	Challenges of Low-Code/No-Code Software Development: A Literature Review
dixit (2022)	1	0,7164	Towards user-centered and legally relevant smart-contract development: A systematic literature review
bhattacharyya (2023)	1	0,6848	Study of deployment of "low code no code" applications toward improving digitization of supply chain management
sundberg (2023)	1	0,4579	Democratizing artificial intelligence: How no-code AI can leverage machine learning operations
jauhar (2023)	1	0,0844	How to use no-code artificial intelligence to predict and minimize the inventory distortions for resilient supply chains
silva (2021)	2	0,2478	Development and Validation of a Descriptive Cognitive Model for Predicting Usability Issues in a Low-Code Development Platform
seiger (2021)	2	0,174	HoloFlows: modelling of processes for the Internet of Things in mixed reality
lethbridge (2021)	2	0,0632	Low-Code Is Often High-Code, So We Must Design Low-Code Platforms to Enable Proper Software Engineering
lago (2021)	2	0,0597	Managing non-trivial internet-of-things systems with conversational assistants: A prototype and a feasibility experiment
tamilselvam (2019)	2	0,0257	A visual programming paradigm for abstract deep learning model development
sufi (2023)	2	0,0113	Algorithms in Low-Code-No-Code for Research Applications: A Practical Review
lei (2022)	3	0,115	Deep learning application on code clone detection: A review of current knowledge
pizzolotto (2021)	3	0,0256	Identifying Compiler and Optimization Level in Binary Code From Multiple Architectures
gupta (2019)	3	0,0168	An Empirical Framework for Code Smell Prediction using Extreme Learning Machine
yu (2022)	3	0,0112	CGFuzzer: A Fuzzing Approach Based on Coverage-Guided Generative Adversarial Networks for Industrial IoT Protocols
kocaman (2021)	3	0,0112	Biomedical Named Entity Recognition at Scale
diEuscio (2022)	4	0,4758	Low-code development and model-driven engineering: Two sides of the same coin?
di sipio (2020)	4	0,1278	Democratizing the development of recommender systems by means of low-code platforms
almonite (2020)	4	0,0785	Towards automating the construction of recommender systems for low-code development platforms
tisi (2019)	4	0,0181	Lowcomote: Training the Next Generation of Experts in Scalable Low-Code Engineering Platforms
al alamin (2021)	5	0,9007	An empirical study of developer discussions on low-code software development challenges
ihirwe (2020)	5	0,1313	Low-code engineering for internet of things: a state of research
kourouklidis (2020)	5	0,1313	Towards a low-code solution for monitoring machine learning model performance
dalibor (2022)	6	1	Generating customized low-code development platforms for digital twins
bano (2022)	6	0,365	Process-aware digital twin cockpit synthesis from event logs
palomba (2018)	7	0,1171	A large-scale empirical study on the lifecycle of code smell co-occurrences
pantiuchina (2018)	7	0,0164	Towards just-in-time refactoring recommenders

Elaboração: Do autor (2024). **Fonte:** GEPHI, 2024.

O Grupo 3 foca na integração de técnicas avançadas de desenvolvimento, como aprendizado profundo, aprendizado de máquina extremo e redes generativas. Os trabalhos apontam melhorias significativas na personalização de soluções e na qualidade do *software*, além de ampliar o acesso a tecnologias de ponta.

No Grupo 4, os estudos refletem sobre a automação e a democratização do desenvolvimento de software. São analisadas plataformas que capacitam usuários não técnicos a participarem ativamente da criação de soluções, integrando técnicas de inteligência artificial e automação.

O Grupo 5 explora tanto os desafios quanto as oportunidades relacionadas ao desenvolvimento *low-code*. Questões como *bugs*, usabilidade e manutenção são contrapostas a benefícios como agilidade, flexibilidade e redução da dependência de equipes altamente técnicas.

O Grupo 6 examina o uso de plataformas *low-code* para desenvolvimento de gêmeos digitais, enfatizando a necessidade de personalização e adaptabilidade dessas soluções. Por fim, o Grupo 7 destaca abordagens para detecção e mitigação de *code smells*, com foco em manter a manutenibilidade e a qualidade em projetos *low-code*.

Os grupos 1, 2 e 4 se destacam por maior proximidade com o foco desta pesquisa. Abordam, respectivamente, os fundamentos do *low/no-code*, sua aplicação em contextos complexos e a capacitação de usuários — elementos centrais para a democratização do

aprendizado de máquina e a formação de desenvolvedores cidadãos. Isso indica que o tema da presente pesquisa está inserido em frentes relevantes da produção científica atual.

DISCUSSÃO DOS RESULTADOS DA PESQUISA

O estudo revelou que as ferramentas *low/no-code* vêm se consolidando como recursos estratégicos para ampliar o acesso ao desenvolvimento de soluções baseadas em inteligência artificial, especialmente entre públicos não técnicos.

A análise de desempenho evidenciou um crescimento expressivo da produção científica sobre o tema, com destaque para o período entre 2014 e 2024 e um pico em 2023, o que aponta para a consolidação do interesse acadêmico em plataformas voltadas à inclusão de novos atores no contexto digital e democratização do aprendizado de máquina, como também observado por Sundberg e Holmström (2023; 2024).

Esses resultados dialogam diretamente com o objetivo central do estudo, que foi investigar como soluções *low-code/no-code* aplicadas ao aprendizado de máquina estão sendo incorporadas às iniciativas de formação de desenvolvedores cidadãos. A análise conceitual reforçou essa conexão ao destacar a recorrência de termos como *citizen developer* e *machine learning*, revelando a articulação desses conceitos na literatura científica recente, conforme apontado por Donthu *et al.* (2021).

A estrutura intelectual mapeada por meio da co-citação demonstrou que a base teórica da área está fortemente ancorada em autores que discutem a integração entre práticas de engenharia de *software* e participação de desenvolvedores cidadãos, como Lethbridge (2021), Atkins (2021) e Hoogsteen e Borgman (2021). Tais obras servem de sustentação conceitual para compreender o papel das plataformas *low/no-code* como facilitadoras da formação de novos perfis de desenvolvedores.

O acoplamento bibliográfico identificou frentes de pesquisa que convergem justamente para os eixos de capacitação, automação, integração com inteligência artificial e inclusão de novos atores no ecossistema tecnológico. Trabalhos como os de El Kamouchi *et al.* (2023) e Hoogsteen e Borgman (2022) ilustram como a literatura recente já reconhece o valor de iniciativas que promovem a formação de desenvolvedores cidadãos como parte da estratégia de inclusão.

Portanto, os resultados obtidos respondem de forma clara à questão de pesquisa ao demonstrar que, embora a integração entre *low/no-code* e aprendizado de máquina ainda esteja

em consolidação, há uma articulação crescente entre essas ferramentas e práticas formativas voltadas à inclusão tecnológica.

O estudo evidencia não apenas o amadurecimento da discussão acadêmica sobre o tema, mas também os caminhos pelos quais essas soluções estão sendo exploradas como suporte à inclusão digital de novos desenvolvedores em ambientes educacionais e corporativos.

CONSIDERAÇÕES FINAIS

O estudo revelou que as ferramentas *low/no-code* têm desempenhado um papel estratégico na democratização do acesso ao aprendizado de máquina e na formação de desenvolvedores cidadãos. A análise bibliométrica demonstrou um crescimento expressivo da produção científica sobre o tema, especialmente a partir de 2023, sinalizando a consolidação dessas plataformas como facilitadoras da transformação digital em diferentes contextos organizacionais.

Os resultados reforçam que essas soluções não apenas aceleram o desenvolvimento tecnológico, mas também ampliam a inclusão de indivíduos sem formação técnica formal no processo de criação de aplicações baseadas em aprendizado de máquina. Essa abertura contribui para a redução de barreiras técnicas, o aumento da autonomia institucional e a promoção de ambientes mais inovadores e colaborativos.

De forma prática, esta pesquisa oferece subsídios para instituições públicas, empresas e organizações educacionais que buscam ampliar o acesso à inteligência artificial por meio de abordagens mais inclusivas de capacitação digital. A identificação de eixos temáticos consolidados e lacunas emergentes pode orientar a formulação de políticas de formação em plataformas *low/no-code*, especialmente voltadas a públicos não técnicos.

Do ponto de vista acadêmico, o estudo contribui para o entendimento das dinâmicas teóricas e temáticas que fundamentam a transformação digital baseada em *low/no-code*. Além disso, aprofunda a compreensão sobre a formação de novos perfis profissionais no contexto da inteligência artificial. Os achados também ampliam a base conceitual necessária para futuras pesquisas sobre inclusão tecnológica e desenvolvimento orientado por usuários.

Como limitação, destaca-se a utilização exclusiva da base de dados Scopus, o que pode restringir a abrangência de determinadas publicações relevantes não indexadas nessa plataforma. Além disso, a análise concentrou-se em dados bibliográficos, sem incluir validação empírica com usuários ou iniciativas institucionais.

Futuros estudos podem combinar abordagens bibliométricas com métodos qualitativos, como entrevistas ou estudos de caso, para aprofundar a compreensão sobre os impactos práticos das plataformas *low/no-code* na formação de desenvolvedores cidadãos.

Recomenda-se que futuras pesquisas aprofundem o desenvolvimento de metodologias educacionais voltadas à formação de desenvolvedores cidadãos, bem como a integração dessas plataformas com tecnologias emergentes. O aprimoramento contínuo dessas soluções poderá fortalecer sua aplicabilidade e ampliar seu impacto na transformação digital e na inclusão tecnológica em larga escala.

REFERÊNCIAS

ALAMIN, M. A. AL; MALAKAR, S.; UDDIN, G.; AFROZ, S.; HAIDER, T. B.; IQBAL, A. An empirical study of developer discussions on low-code software development challenges. In: IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), 2021, [S. l.]. **Anais [...]**. IEEE, 2021. p. 46–57. DOI: <https://doi.org/10.1109/MSR52588.2021.00018>.

ARIA, M.; CUCCURULLO, C. bibliometrix: An R-tool for comprehensive science mapping analysis. **Journal of Informetrics**, v. 11, n. 4, p. 959–975, 2017. DOI: <https://doi.org/10.1016/j.joi.2017.08.007>.

ASHRAF, A. R.; SOMOGYI-VÉGH, A.; MERCZEL, S.; GYIMESI, N.; FITTLER, A. Leveraging code-free deep learning for pill recognition in clinical settings: a multicenter, real-world study of performance across multiple platforms. **Artificial Intelligence in Medicine**, [S. l.], v. 150, 2024. DOI: 10.1016/j.artmed.2024.102844.

ATKINS, B. The most disruptive trend of 2021: no code / low code. **Forbes**, [S. l.], 2021. Disponível em: <https://www.forbes.com/sites/betsyatkins/2020/11/24/the-most-disruptive-trend-of-2021-no-code--low-code/> Acesso em: 30 maio 2024.

AUREA. AUREA BPM PLATFORM OVERVIEW. AUSTIN: AUREA SOFTWARE INC., 2023. **Página Inicial**. Disponível em: <https://www.tecna.pl/en/oferta/platforma-aurea-bpm>. Acesso em: 5 abr. 2025.

BARRICELLI, B. R.; CASSANO, F.; FOGLI, D.; PICCINNO, A. End-user development, end-user programming and end-user software engineering: a systematic mapping study. **Journal of Systems and Software**, [S. l.], v. 149, p. 101–137, 2019. DOI: 10.1016/j.jss.2018.11.041

BASTIAN, M.; HEYMANN, S.; JACOMY, M. Gephi: An *open source* software for exploring and manipulating networks. In: INTERNATIONAL AAAI CONFERENCE ON WEBLOGS AND SOCIAL MEDIA, 2009, San Jose. Proceedings. Menlo Park: AAAI Press, 2009. p. 361–362. DOI: <https://doi.org/10.1609/icwsm.v3i1.13937>.

BIBLIOMETRIX. *COMPREHENSIVE SCIENCE MAPPING ANALYSIS*. **Página Inicial**. [s.l.], 2025. Disponível em: <https://www.bibliometrix.org/home/>. Acesso em: 12 maio 2025.

BOCK, A. C.; FRANK, U. In search of the essence of low-code: an exploratory study of seven development platforms. **ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)**, [S. l.], p. 57–66, 2021. DOI: 10.1109/MODELS-C53483.2021.00016.

BOCK, A. C.; FRANK, U. Low-code platform. **Business and Information Systems Engineering**, [S. l.], v. 63, n. 6, p. 733–740, 2021. DOI: 10.1007/s12599-021-00726-8.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, 2001. DOI: <https://doi.org/10.1023/A:1010933404324>. Disponível em: <https://link.springer.com/article/10.1023/A:1010933404324>. Acesso em: 13 jul. 2025.

BUCAIONI, A.; CICHETTI, A.; CICOZZI, F. Modelling in low-code development: a multi-vocal systematic review. **Software and Systems Modeling**, [S. l.], v. 21, n. 5, p. 1959–1981, 2022. DOI: 10.1007/s10270-021-00964-0.

CURTY, S.; HÄRER, F.; FILL, H.-G. Design of blockchain-based applications using model-driven engineering and low-code/no-code platforms: a structured literature review. **Software and Systems Modeling**, [S. l.], v. 22, n. 6, p. 1857–1895, 2023. DOI: 10.1007/s10270-023-01109-1.

DANIEL, G.; CABOT, J.; DERUELLE, L.; DERRAS, M. Xatkit: a multimodal low-code chatbot development framework. **IEEE Access**, [S. l.], v. 8, p. 15332–15346, 2020. DOI: 10.1109/aACCESS.2020.2966919.

DI RUSCIO, D.; KOLOVOS, D.; DE LARA, J.; PIERANTONIO, A.; TISI, M.; WIMMER, M. Low-code development and model-driven engineering: two sides of the same coin? **Software and Systems Modeling**, [S. l.], v. 21, n. 2, p. 437–446, 2022. DOI: 10.1007/s10270-021-00970-2.

DONTHU, N.; KUMAR, S.; MUKHERJEE, D.; PANDEY, N.; LIM, W. M. How to conduct a bibliometric analysis: An overview and guidelines. **Journal of Business Research**, v. 133, p. 285–296, 2021. DOI: <https://doi.org/10.1016/j.jbusres.2021.04.070>.

EL KAMOUCI, H.; KISSI, M.; EL BEGGAR, O. Low-code/no-code development: a systematic literature review. Proceedings - SITA 2023: 2023 14th **International Conference on Intelligent Systems: Theories and Applications**, [S. l.], 2023. DOI: 10.1109/SITA60746.2023.10373712.

FAVRE, J.-M.; DE OLIVEIRA, R. A.; SOTTET, J.-S.; QUAST, M. In search of the essence of no-code - elements of data modeling. Proceedings - 2023 **ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion**, [S. l.], p. 878–887, 2023. DOI: 10.1109/MODELS-C59198.2023.00138.

GEPHI. *THE OPEN GRAPH VIZ PLATFORM*. 2024. Disponível em: <https://gephi.org/>. Acesso em: 12 maio 2025.

HIRZEL, M. Low-code programming models. **Communications of the ACM**, [S. l.], v. 66, n. 10, p. 76–85, 2023. DOI: 10.1145/3587691.

HOOGSTEEEN, D.; BORGMAN, H. Empower the workforce, empower the company? Citizen development adoption. Proceedings of the **Annual Hawaii International Conference on System Sciences**, [S. l.], p. 4717–4726, 2022. DOI: 10.24251/HICSS.2022.575.

HURLBURT, G. Low-code, no-code, what's under the hood? **IT Professional**, [S. l.], v. 23, n. 6, p. 4–7, 2021. DOI: 10.1109/MITP.2021.3123415.

IBM Cloud Education. **Low-code vs no-code: what's the difference?** [S. l.], 2022. Disponível em: <https://www.ibm.com/think/topics/low-code-vs-no-code>. Acesso em: 30 maio 2024.

KASS, S.; STRAHRINGER, S.; WESTNER, M. Drivers and inhibitors of low code development platform adoption. **Proceedings - 2022 IEEE 24th Conference on Business Informatics**, CBI 2022, [S. l.], v. 1, p. 196–205, 2022. DOI: 10.1109/CBI54897.2022.00028.

LETHBRIDGE, T. C. Low-code is often high-code, so we must design low-code platforms to enable proper software engineering. In: **INTERNATIONAL SYMPOSIUM ON LEVERAGING APPLICATIONS OF FORMAL METHODS**, 2021, Heraklion. Proceedings. Cham: Springer, 2021. p. 202–212. DOI: 10.1007/978-3-030-89159-6_14.

LIEBERMAN, H.; PATERNO, F.; KLANN, M.; WULF, V. End-user development: an emerging paradigm. In: **END USER DEVELOPMENT. HUMAN-COMPUTER INTERACTION SERIES**, [S. l.], v. 9, Springer Netherlands, 2006, p. 1–8. DOI: 10.1007/1-4020-5386-X_1.

LIU, C.; DOWNING, C. Using text analytics ai insights in microsoft power bi desktop to score sentiments, extract key phrases, and discover unstructured data patterns. **Journal of Information Systems Education**, [S. l.], v. 35, n. 1, p. 48–55, 2024. DOI: 10.62273/PKER1800.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arxiv.org**, [S. l.], 2013. Disponível em: <https://arxiv.org/abs/1301.3781>. Acesso em: 30 maio 2024.

OPENAI; ACHIAM, J.; ADLER, S.; AGARWAL, S.; AHMAD, L.; AKKAYA, I.; et al. GPT-4 TECHNICAL REPORT. **arxiv.org**, [S. l.], 2024. Disponível em: <https://arxiv.org/abs/2303.08774>. Acesso em: 30 maio 2024.

OVEREEM, M.; JANSEN, S. ROPOSING a framework for impact analysis for low-code development platforms. In: **COMPANION PROCEEDINGS - 24TH INTERNATIONAL CONFERENCE ON MODEL-DRIVEN ENGINEERING LANGUAGES AND SYSTEMS, MODELS-C**. 2021, p. 88–97. DOI: 10.1109/models-c53483.2021.00020.

PICEK, R. Low-code/no-code platforms and modern erp systems. In: **PROCEEDINGS - 2023 9TH INTERNATIONAL CONFERENCE ON INFORMATION MANAGEMENT, ICIM 2023**, [S. l.], p. 44–49, 2023. DOI: 10.1109/ICIM58774.2023.00014. DOI: <https://doi.org/10.1109/ICIM58774.2023.00014>.

RAFFEL, C.; SHAZEER, N.; ROBERTS, A.; LEE, K.; NARANG, S.; MATENA, M. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. **arxiv.org**, [S. l.], 2023. DOI: <https://doi.org/10.48550/arXiv.1910.10683>.

RAGHAVENDRAN, K. R.; ELRAGAL, A. Low-code machine learning platforms: a fastlane to digitalization. **Informatics**, [S. l.], v. 10, n. 2, 2023. DOI: 10.3390/informatics10020050.

RAZAK, S. F. A.; ERNN, Y. P.; YUSSOFF, F. I.; BUKAR, U. A.; YOGARAYAN, S. Enhancing business efficiency through low-code/no-code technology adoption: insights from an extended utaut model. **Journal of Human, Earth, and Future**, [S. l.], v. 5, n. 1, p. 85–99, 2024. DOI: 10.28991/HEF-2024-05-01-07.

RICHARDSON, C.; RYMER, J. New development platforms emerge for customer-facing applications. [S. l.]: **Forrester**, 2014. Disponível em: <https://www.forrester.com/report/New-Development-Platforms-Emerge-For-CustomerFacing-Applications/RES113411>. Acesso em: 30 mai. 2024.

RICHARDSON, C.; RYMER, J. Vendor landscape: the fractured, fertile terrain of low-code application platforms. **Forrester**, [S. l.], 2016. Disponível em: <https://www.forrester.com/report/Vendor-Landscape-The-Fractured-Fertile-Terrain-Of-LowCode-Application-Platforms/RES122549>. Acesso em: 30 mai. 2024.

ROKIS, K.; KIRIKOVA, M. Challenges of low-code/no-code software development: a literature review. **Lecture Notes in Business Information Processing**, [S. l.], v. 462, p. 3–17, 2022. DOI: 10.1007/978-3-031-16947-2_1.

SAHAY, A.; INDAMUTSA, A.; DI RUSCIO, D.; PIERANTONIO, A. Supporting the understanding and comparison of low-code development platforms. **Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications**, 2020, [S. l.], p. 171–178, 2020. DOI: 10.1109/SEAA51224.2020.00036.

SANCHIS, R.; GARCÍA-PERALES, Ó.; FRAILE, F.; POLER, R. Low-code as enabler of digital transformation in manufacturing industry. **Applied Sciences (Switzerland)**, [S. l.], v. 10, n. 1, 2020. DOI: 10.3390/app10010012.

SILVA, C.; VIEIRA, J.; CAMPOS, J. C.; COUTO, R.; RIBEIRO, A. N. Development and validation of a descriptive cognitive model for predicting usability issues in a low-code development platform. **Human Factors**, [S. l.], v. 63, n. 6, p. 1012–1032, 2020. DOI: 10.1177/0018720820920429.

SOERGEL, D. The Art and Architecture Thesaurus (AAT): a critical appraisal. **Visual Resources**, v. 10, n. 4, p. 369–400, 1995. DOI: [10.1080/01973762.1995.9658306](https://doi.org/10.1080/01973762.1995.9658306).

SUNDBERG, L.; HOLMSTRÖM, J. Democratizing artificial intelligence: how no-code ai can leverage machine learning operations. **Business Horizons**, [S. l.], v. 66, n. 6, p. 777–788, 2023. DOI: 10.1016/j.bushor.2023.04.003.

SUNDBERG, L.; HOLMSTRÖM, J. Using no-code ai to teach machine learning in higher education. **Journal of Information Systems Education**, [S. l.], v. 35, n. 1, p. 56–66, 2024. DOI: 10.62273/CYPL2902.

VARAJÃO, J. Software development in disruptive times. **Communications of the ACM**, [S. l.], v. 64, n. 10, p. 32–35, 2021. DOI: [10.1145/3453932](https://doi.org/10.1145/3453932)

VF-OS PROJECT. VIRTUAL FACTORY OPEN OPERATING SYSTEM – **Final Report**. Bruxelas: Comissão Europeia, 2020. Disponível em: <https://innovation-radar.ec.europa.eu/innovation/36748>. Acesso em: 5 abr. 2025.

VAN ECK, N. J.; WALTMAN, L. Software survey: VOSviewer, a computer program for bibliometric mapping. **Scientometrics**, v. 84, n. 2, p. 523–538, 2010. DOI: <https://doi.org/10.1007/s11192-009-0146-3>.

VOSVIEWER. *VOSVIEWER – VISUALIZING SCIENTIFIC LANDSCAPES*. **Página Inicial**. [s.l], 2024. Disponível em: <https://www.vosviewer.com/>. Acesso em: 12 maio 2025.

XATKIT. **XATKIT: CONVERSATIONAL PLATFORM FOR CHATBOTS**. [s.l.], 2023. Disponível em: <https://xatkit.com>. Acesso em: 5 abr. 2025.

WASZKOWSKI, R. Low-code platform for automating business processes in manufacturing. **IFAC-PapersOnLine**, [S. l.], v. 52, n. 10, p. 376–381, 2019. DOI: 10.1016/j.ifacol.2019.10.060.

WOO, M. The rise of no/low code software development—no experience needed? **Engineering**, [S. l.], v. 6, n. 9, p. 960–961, 2020. DOI: 10.1016/J.ENG.2020.07.007.