



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
SISTEMAS DE INFORMAÇÃO

JHONATA LIMA DE SOUZA

**A Contribuição do Low Code no Âmbito Educacional: Um
Mapeamento Sistemático da Literatura**

Recife
2022

JHONATA LIMA DE SOUZA

**A Contribuição do Low Code no Âmbito Educacional: Um
Mapeamento Sistemático da Literatura**

Trabalho apresentado ao Programa de Graduação em Sistemas de informação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientadora: Prof. Simone C. dos Santos

Recife
2022

JHONATA LIMA DE SOUZA

**A Contribuição do Low Code no Âmbito Educacional: Um
Mapeamento Sistemático da Literatura**

Trabalho apresentado ao Programa de Graduação em Sistemas de informação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Recife, 10 de maio de 2022

BANCA EXAMINADORA

Prof. Simone C. dos Santos (Orientadora)
UNIVERSIDADE FEDERAL DE PERNAMBUCO

Prof. Filipe Calegário (2º membro da banca)
UNIVERSIDADE FEDERAL DE PERNAMBUCO

Dedico este trabalho aos meus pais, guerreiros que mesmo diante de toda dificuldade me ensinaram desde cedo a magia da curiosidade.

AGRADECIMENTOS

Agradeço imensamente a minha família por todo o apoio. Em especial meus pais, Janeide Porcina de Lima e Elias Ferreira de Souza, pelo apoio e todo ensinamento em diferentes momentos da vida. Ao meu irmão Everton Lima de Souza pelo companheirismo e todas as experiências durante esse momento.

A minha namorada Maria Eduarda por tornar essa existência mais suave e sutil, assim como o apoio emocional dado durante este trabalho.

Agradeço aos meus amigos, que tornam toda a experiência da graduação mais suave e divertida, assim como proporcionaram trocas de informação e conhecimentos valiosíssimos. Em especial para minha amiga antropóloga Maria Dairly, cuja troca de informações me ajudou a fortalecer a curiosidade e o lado humano das relações tecnológicas. Aos meus amigos de graduação Bernardo, Gabriel e Vitor, pelas trocas de informações e apoio nos momentos difíceis.

Um agradecimento especial para minha orientadora Simone Cristiane dos Santos que me deu todo o apoio e a oportunidade de pesquisar um tema relevante e muito significativo para mim e que, assim como todo o pesquisador/cientista (profissão mais importante da humanidade), merece todos os aplausos por contribuir com o progresso do ensino e da ciência brasileira.

"Ensinar não é transferir conhecimento, mas criar as possibilidades para a sua própria produção ou a sua construção." (Paulo Freire)

RESUMO

A educação na área de tecnologia da informação e comunicação (TIC) muda com uma grande velocidade. Isso devido ao fato da mesma estar na linha de frente da inovação. A partir disso, foi percebido que o aprendizado ativo, método que coloca o aluno como protagonista do processo de aprendizagem, já vem sendo utilizado nas universidades e escolas de tecnologia. Muitos métodos utilizam do aprendizado ativo para simular o ambiente de trabalho, sendo necessário o uso de ferramentas que auxiliem os alunos nessa jornada. As ferramentas Low Code possuem um grande apelo pela facilidade de desenvolvimento, integração e implantação, sendo a alta abstração de código o seu principal artifício. Por consequência disso, organizações começaram a adotar sistematicamente esta tecnologia em seus portfólios. Sendo esperado grande adoção dessa tecnologia por grande parte do mercado de software é necessário que os novos profissionais da área sejam educados de maneira que possam acompanhar essa tecnologia. Com essa motivação, a seguinte questão central de pesquisa foi definida: QC – “*Como o Low Code está sendo usado no âmbito educacional?*”. Para responder essa questão, foi conduzido um estudo de mapeamento sistemático sobre o uso do Low Code no âmbito educacional, de forma que possa ser elucidado seu uso nos diferentes níveis educacionais, como é utilizado, quais benefícios para os futuros profissionais, vantagens e desvantagens do uso do Low Code. A partir do método de mapeamento, trinta e nove (39) estudos foram selecionados em diferentes mecanismos de busca. Foi observado que o Low Code possui utilização não só no ensino superior, mas também, no ensino médio, cursos de treinamento, inovação e até no uso de tecnologia aplicada à agricultura. O uso do Low Code foi coerente com a sua proposta para o desenvolvimento de aplicações, mas vários outros usos foram identificados, como apoio a cursos de treinamento profissional, desenvolvimento de algoritmos de IA e inovação. O uso das ferramentas Low Code foi associado a vantagens como produtividade no desenvolvimento de software e facilidade em alcançar objetivos educacionais relacionados a área de TIC, atribuindo competitividade para os novos profissionais. A Ferramenta também se mostrou promissora na oportunidade da exploração de conceitos do mundo real no desenvolvimento de software, com alguns desafios, como limitações da própria ferramenta e falta de padrões de licença nas diferentes ferramentas. Por fim, esse estudo conseguiu responder à pergunta proposta e se espera que novos trabalhos aplicados com o Low Code sejam conduzidos na área da educação, a partir dos resultados encontrados.

Palavras-chave: Educação, Low Code, Mapeamento Sistemático da Literatura.

ABSTRACT

Education in the field of information and communication technology (ICT) changes at a high speed. This is due to the fact that it is at the forefront of innovation. From this, it was noticed that active learning, a method that places the student as the protagonist of the learning process, is being used in universities and technology schools. Many methods use active learning to simulate the work environment, requiring the use of tools that help students on this journey. Low Code tools have a great appeal for the ease of development, integration, and deployment, with high abstraction of code being their main artifice. As a result, organizations began to systematically adopt this technology in their portfolios. As it is expected that this technology will be widely adopted by a large part of the software market, new professionals in the area must be educated with this tool so that they can follow this technology. With this motivation, the following central research question was defined: QC – *“How is the Low Code being used in the educational field?”*. To answer this question a systematic mapping study of the use of Low Code in the educational field was carried out, so that its use at different educational levels can be elucidated, how it is used, what benefits for future professionals, advantages and disadvantages of using Low Code. From the mapping method, thirty-nine (39) studies were selected based on the mapping method, in different search engines. It was observed that the Low Code is used not only in the superior education, but also in high school, training courses, innovation, and even in the use of technology applied to agriculture. The use of Low Code was consistent with its proposal, for the development of applications, but several other uses were identified, such as support for professional training courses, AI algorithms, and innovation. The use of Low Code tools was associated with advantages such as productivity in software development and ease of achieving educational goals related to the ICT area, attributing competitiveness to new professionals. The platform also showed promise as an opportunity to explore real-world concepts in software development, with some challenges, such as limitations of the tool itself and lack of license standards on different platforms. Finally, this study was able to answer the proposed question and it is expected that new works applied with the Low Code will be conducted in the area of education, based on the results found.

Keywords: Education, Low Code, Systematic Mapping.

LISTA DE ILUSTRAÇÕES

Figura 1 — Editor visual da ferramenta Low Code Outsystems.	22
Figura 2 — Gerenciamento de ambientes em nuvem da ferramenta Low Code Outsystems .	23
Figura 3 — Processo de versionamento e <i>deployment</i> facilitado das aplicações da ferramenta Low Code Outsystems	23
Figura 4 — Etapas do método proposto por Kitchenham (2007).....	25
Figura 5 — Estudos por ano.	32
Figura 6 — Estudos por base de dados e por ano.	32
Figura 7 — Distribuição dos níveis educacionais do mapeamento.	33
Figura 8 — Distribuição das áreas do conhecimento.	36
Figura 9 — Gráfico mostrando o número de estudos identificados para cada uso da ferramenta.	38
Figura 10 — Ferramentas Low Code utilizadas nos estudos.	39
Figura 11 — Plano de curso para retreino de profissionais na área de TIC.	44

LISTA DE TABELAS

Tabela 1 — Resultados encontrados nas principais etapas do processo de seleção por base de dados.	30
Tabela 2 — Resultados finais dos estudos selecionados para o mapeamento sistemático	31
Tabela 3 — Níveis educacionais identificados nos estudos do mapeamento.....	34
Tabela 4 — Relação entre curso e/ou disciplina e estudos. (Continua)	35
Tabela 5 — Estudos que responderam como as ferramentas Low Code são utilizadas.	38
Tabela 6 — Relação dos estudos na utilização do Low Code nos diferentes níveis educacionais.....	40
Tabela 7 — Benefícios e vantagens do aprendizado do Low Code para a formação profissional.	42
Tabela 8 — Desafios relacionados ao uso do Low Code no âmbito educacional.	46
Tabela 9 — Oportunidades relacionadas ao uso do Low Code no âmbito educacional.	50

LISTA DE ABREVIATURAS E SIGLAS

AMCIS	Americas Conference on Information Systems
BDD	Desenvolvimento Orientado Por Comportamento
ECIS	European Conference on Information Systems
HICSS	Hawaii International Conference on System Sciences
ICIS	International Conference on Information Systems
PACIS	Pacific Asia Conference on Information Systems
PBL	Aprendizagem Baseada Em Problema
PrBL	Aprendizagem Baseada Em Projeto
STEAM	Ciência, Tecnologia, Engenharia Artes e Matemática
TDD	Desenvolvimento Guiado Por Testes
TIC	Tecnologia da Informação e Comunicação

SUMÁRIO

1	INTRODUÇÃO	12
1.1	CONTEXTO E MOTIVAÇÃO	12
1.2	JUSTIFICATIVA	14
1.3	OBJETIVOS DA PESQUISA	15
1.4	ESTRUTURA DE TRABALHO.....	16
2	PRINCIPAIS REFERENCIAIS TEÓRICOS	17
2.1	EDUCAÇÃO NA CIÊNCIA DA COMPUTAÇÃO	17
2.1.1	Métodos ágeis na educação	18
2.1.2	Aprendizagem baseada em Projetos (PrBL)	19
2.1.3	Aprendizagem baseada em Problema (PBL)	20
2.2	LOW CODE	21
3	METODOLOGIA	25
3.1	PERGUNTAS DA PESQUISA.....	26
3.2	BUSCA E SELEÇÃO DOS ESTUDOS.....	27
4	RESULTADOS	31
4.1	DADOS GERAIS	31
4.2	QUESTÃO 1 - QUAIS DISCIPLINAS E/OU NÍVEIS EDUCACIONAIS ABORDAM O LOW CODE?	33
4.2.1	Qual o nível educacional?	33
4.2.2	Quais cursos, disciplinas e áreas do conhecimento?	34
4.3	QUESTÃO 2 - COMO O LOW CODE TEM SIDO UTILIZADO?	37
4.4	QUESTÃO 3 - POR QUE APRENDER LOW CODE (BENEFÍCIOS E VANTAGENS PARA A FORMAÇÃO PROFISSIONAL)?	42
4.5	QUESTÃO 4 - QUAIS SÃO OS PRINCIPAIS DESAFIOS, LIMITAÇÕES, DESVANTAGENS E OPORTUNIDADES RELACIONADOS AO USO DE LOW CODE EM EDUCAÇÃO?	46
4.5.1	Quais são os desafios, limitações e desvantagens?	46
4.5.2	Quais são as oportunidades?	49
5	CONCLUSÃO	52
5.1	RETOMANDO AS QUESTÕES DE PESQUISA.....	52
5.2	LIMITAÇÕES E AMEAÇAS DA PESQUISA	54
5.3	TRABALHOS FUTUROS	54
	REFERÊNCIAS	55
	APÊNDICE A — Estudos primários identificados nesse mapeamento.	59
	APÊNDICE B — Índice de responsividade dos estudos para as questões de pesquisa.....	63

1 INTRODUÇÃO

1.1 CONTEXTO E MOTIVAÇÃO

A educação em áreas da computação é algo extremamente dinâmico. Isso se dá por consequência dessa área estar à frente nas inovações e tecnologias atuais. Por consequência disso, as mudanças nessa área tendem a ser rápidas e costumam afetar tanto as empresas do ramo, como o ambiente educacional (OLIVEIRA, 2013). Portanto, um método de ensino enrijecido e com bastante conteúdo vem sendo combatido, de modo a evitar o desinteresse na área e a formação profissional defasada (OLIVEIRA, 2013).

O ideal é que processo de aprendizagem na computação seja bastante fluído e acompanhe os métodos e inovações nessa área. No geral, para combater o enrijecimento do ensino esses métodos utilizam o aprendizado ativo, onde o aluno deve buscar o conhecimento de forma constante, a partir de diversas fontes, sendo o professor apenas um facilitador, quem vai trazer problemas, métodos e direções para a correta construção do conhecimento e do estímulo ao pensamento crítico (DIESEL, 2017). Também podendo ser dito que, essas metodologias utilizam inovações e ferramentas para simular o ambiente de trabalho para que seja possível melhorar e reforçar o aprendizado, a empregabilidade e o treinamento dos futuros profissionais. Dessa forma, ferramentas para atingir esses objetivos são necessárias (DEWI, 2014; DIESEL, 2017).

O Low Code é uma ferramenta que utiliza de alta abstração de código para geração de aplicações. Com apenas um clique, uma aplicação ou conjunto de aplicações pode, instantaneamente, ser integrado a códigos customizados, ser implantado ao ambiente de produção e isso tudo manipulado através de servidores em nuvem. Com diversas facilidades, o Low Code vem ganhando espaço no mercado de trabalho e já é parte de relatórios anuais do Gartner, incluindo previsões que vêm se concluindo a cada ano. Um exemplo disso é que o instituto Gartner prevê o uso de ferramentas Low Code em 50% das empresas de médio e grande porte em 2023 (VINCENT, 2020). O Gartner ainda aponta que, em 2024, a maioria dos produtos e serviços tecnológicos serão construídos por profissionais fora de TI e uma das ferramentas que ajudará esse cenário se tornar realidade é o desenvolvimento de software Low Code (RIMOL, 2021).

As plataformas No Code também são bastante comentadas na atualidade (ATKINS, 2020). No contexto geral, essa ferramenta possui um objetivo semelhante ao Low Code, que é

trazer facilidade no desenvolvimento através de um alto nível de abstração, porém é um tipo de ferramenta fechada e não é possível customizar aplicações através de códigos, sendo apenas editores visuais de aplicações, diferente das ferramentas Low Code, que são abertas ao código tradicional através de integrações e extensões (MEIJERS, s.d.; OUTSYSTEMS, 2022).

Com a presente mudança de perspectiva no mercado e com uma ferramenta que integra editores visuais ao desenvolvimento altamente produtivo e com possibilidade da customização e integração de código tradicional, o Low Code se torna uma ferramenta chave para o ensino nas áreas de Tecnologia da Informação e Comunicação (TIC). Assim, o Low Code pode ser integrado às disciplinas e cursos diversos, tendo diferentes usos e propósitos, incluindo conceitos utilizados no desenvolvimento de software tradicional visto de uma forma totalmente diferente.

1.2 JUSTIFICATIVA

A tecnologia Low Code já faz parte do mercado de trabalho e também apresenta uma grande ascensão na sua adoção como ferramenta principal no desenvolvimento de software (VINCENT, 2021). A partir desse cenário, o aprendizado nessa tecnologia é de suma importância, para que profissionais, dentro ou fora da área de TI, possam entender e se adequar para que possam participar dessa nova oportunidade. Apesar das ferramentas Low Code já fazerem parte do ensino no desenvolvimento de software, ainda é escasso o número de trabalhos na área da educação que investiguem sua adoção. Numa busca ad hoc da literatura, nenhum resultado sistemático e consolidado foi encontrado dentro deste tema, o que motivou ainda mais a realização desta pesquisa.

1.3 OBJETIVOS DA PESQUISA

A partir da escassez de trabalhos nessa área, essa pesquisa tem como objetivo desenvolver um mapeamento sistemático para identificar o uso de tecnologias Low Code no âmbito educacional.

Dentro desse contexto os objetivos específicos são:

- Enumerar quais ambientes educacionais estão utilizando essa tecnologia.
- Identificar a contribuição do desenvolvimento de software Low-Code para a educação nos diferentes contextos, disciplinas e níveis educacionais.
- Identificar as vantagens profissionais e educacionais em utilizar o low-code como ferramenta educacional.
- Identificar quais os principais desafios e oportunidades do Low Code como ferramenta na educação.

1.4 ESTUTRURA DE TRABALHO

A organização do trabalho se dá pela seguinte forma:

- Capítulo 1 – Introdução e motivação, apresentando um contexto geral sobre o presente trabalho.
- Capítulo 2 – Principais Referenciais teóricos, descrevendo os temas principais relacionados à educação e às ferramentas Low Code.
- Capítulo 3 – Metodologia, definindo os métodos utilizados no mapeamento sistemático.
- Capítulo 4 – Resultados, apresentando as respostas obtidas através de estudos identificados pelo mapeamento sistemático.
- Capítulo 5 – Conclusão, discutindo os resultados obtidos, limitações, ameaças e a possibilidade de trabalhos futuros.

2 PRINCIPAIS REFERENCIAIS TEÓRICOS

2.1 EDUCAÇÃO NA CIÊNCIA DA COMPUTAÇÃO

Possivelmente, a área da TIC é a área mais dinâmica frente às mudanças e avanços tecnológicos. O que impacta diretamente o mercado de trabalho e, por consequência, toda a estrutura de aprendizado precisa ser constantemente adaptada face ao grande impacto causado pelas mudanças rápidas nesta área, como comentado por Dos Santos (2022) e Oliveira (2013).

De certo modo, esta preocupação não é recente, Tucker em 1996 já estudava um apurado de direções estratégicas na educação da computação, citando diversas estratégias, desafios e oportunidades nessa área (TUCKER, 1996). Para lidar com o dinamismo entre a constante evolução da tecnologia e a dinâmica do ensino, foi publicado pela IEEE/ACM um currículo com recomendações de diversos tipos de conteúdo, entre eles o envolvimento de práticas para resolução de problemas, incremento de habilidades pessoais e interpessoais e trabalho em equipe (DRAFT, 2013). Dos Santos (2022) também comenta que este currículo ainda permanece muito técnico, causando diversos problemas com grandes quantidades de assuntos e pouca prática, pouca motivação, pouco engajamento e alto índice de frustração.

Para sanar esse problema, um paradigma de ensino tem sido reportado por alguns pesquisadores: o paradigma de ensino ativo (OLIVEIRA, 2013; DEWI E MUNIANDY, 2014; DIESEL, 2017). Diferente do ensino passivo, onde o aprendizado é apenas transmitido do professor para o aluno, o ensino ativo envolve o aluno no processo de aprendizado dos conceitos básicos de algumas disciplinas, seja através do estudo off-line, online ou das discussões em grupos, sendo o professor apenas um facilitador deste conhecimento (BRAME, 2016).

Com o passar do tempo, pesquisadores observaram que outros métodos se basearam no conceito do aprendizado ativo para incorporar diversas metodologias para o fortalecimento do aprendizado e a construção de habilidades necessárias a partir de simulações. Ao investigar a literatura científica é possível identificar que algumas pesquisas utilizam de tecnologias emergentes, especificamente o Low Code, em conjunto com essas metodologias para amplificar os benefícios e auxiliar no objetivo educacional das disciplinas em que os métodos e o Low Code foram aplicados. As abordagens proeminentes nesse cenário são os métodos ágeis, aprendizagem baseada em projeto e aprendizagem baseada em problema (COCCO,

2006; OLIVEIRA, 2013; DEWI E MUNIANDY, 2014; DIESEL, 2017; MEW, 2018; FERNANDES, 2020; METRÔLHO, 2022; DOS SANTOS, 2022).

2.1.1 Métodos ágeis na educação

Os métodos ágeis se baseiam em valores e princípios trazidos e organizados no manifesto ágil, com o objetivo de melhorar os processos de desenvolvimento de software tradicionais e lineares, que não pareciam adequados para enfrentar o mundo real em que requisitos e soluções evoluem continuamente (BECK *et al.*, 2001). Segundo Cockburn e Highsmith (2001), a maior implicação do uso do método ágil é que ele possui um grande foco no fator humano, amplificando talentos e habilidades dos indivíduos, o que ocasiona uma maior interação e excelente comunicação, fazendo disso um bom atributo para alcançar objetivos comuns. Para fazer jus a esses atributos, o manifesto ágil cita que um dos mais importantes princípios é referente à construção de indivíduos motivados, com ambiente e suporte necessário para desenvolver o seu trabalho (BECK *et al.*, 2001).

Desde o primeiro momento em que os métodos ágeis se tornaram úteis nos times e projetos de desenvolvimento de software, que algumas pesquisas apareceram no sentido de adaptar esta metodologia para a sala de aula (DEWI E MUNIANDY, 2014). E este fato se iniciou a partir da mudança de paradigma do método de aprendizado, onde foi possível notar aumento do interesse no aprendizado ativo.

O primeiro relato da introdução do método ágil na sala de aula foi descrito por Alfonso e Botia (2005). O método foi introduzido para fazer parte de uma disciplina de engenharia de software, onde o professor gerenciou alunos em um projeto de software real. Já em 2020, foi observado o método Scrum sendo utilizado juntamente a uma ferramenta Low Code para dar suporte a uma disciplina de engenharia de software (METRÔLHO, 2020).

Em uma revisão bibliográfica feita por Salza (2019), foi visto que os métodos ágeis foram utilizados com efetividade para o ensino de engenharia de software, tendo sua maioria de estudos utilizado o método chamado aprendizagem baseada em projeto (PrBL), que será descrito no próximo tópico deste trabalho. No geral, esse método utilizado na educação de engenharia de software serviu tanto como uma técnica educacional, para ser utilizada pelo professor, quanto como o aprendizado da técnica em si, para os alunos. Também foi possível verificar que este modelo permitiu que o conhecimento esperado fosse fornecido de forma

eficiente, adquirido gradativamente, avaliado e, com isso, o processo de aprendizagem fosse efetivamente conduzido (ALFONSO E BOTIA, 2005; SALZA, 2019).

2.1.2 Aprendizagem baseada em Projetos (PrBL)

O método PrBL é uma forma de aprendizado centrado no aluno, ou seja, uma forma de aprendizado ativo. Esta metodologia possui três princípios: todo aprendizado é contexto específico, alunos são ativos no processo de aprendizagem e que os objetivos são alcançáveis através das interações sociais e da partilha de conhecimento (COCCO, 2006). Este método ainda é comentado por outros pesquisadores (BLUMENFELD *et al.*, 2000; KOKOTSAKI, 2016) como um método proposto para possibilitar o uso da simulação em sala de aula. Com isso, estimulando o pensamento crítico e a construção do conhecimento por meio do trabalho em grupo, investigação de problemas reais do dia a dia, levantamento de perguntas e de questões refinadas, design e desenvolvimento de investigações e coleta, análise e divulgação de dados.

Este método também possui pontos em comum com o método de Aprendizagem baseada em Problema (PBL), onde ambos focam no aprendizado ativo do indivíduo, na construção de conhecimento, colaboração e simulação do mundo real, ou profissional. A diferença entre eles é que, o PBL possui foco principal no processo de aprendizado durante o processo de identificação do problema e da teorização da solução. Já o PrBL precisa estender esse conceito através de projetos, com problemas e escopos já definidos, culminando em um produto final (WOODS, 1994; KOKOTSAKI, 2016).

Kokotsaki (2016) também fala sobre o uso do PrBL em diversos níveis educacionais. Variando da escola primária até o ensino superior. Os benefícios citados indicam no geral um aumento do engajamento em grupo, do aumento da motivação e da atitude positiva nos alunos. Mioduser (2006) analisa esse método para superar as barreiras do ensino tecnológico no ensino médio, mostrando um modelo eficiente para um alto nível de aprendizado, sendo possível o aprendizado da educação formal como esperado, não só do aprendizado do projeto feito pelo grupo de alunos, uma vez que este conhecimento deve ser compartilhado com toda a turma. Já Lasauskiene (2015) discute os pontos positivos e negativos vistos pelos professores do ensino superior na inserção deste método, sendo possível identificar que houve desenvolvimento das competências dos alunos, assim como a autorrealização e aprimoramento profissional dos professores e alunos.

2.1.3 Aprendizagem baseada em Problema (PBL)

Segundo Barrows (1996) o PBL é um método educacional que teve sua origem baseada em um problema observado nas escolas de medicina: a insatisfação dos alunos por consequência da alta carga de assuntos para aprender, sendo muito deles com pouca relevância para o exercício da medicina. O autor segue discutindo que o método tradicional de ensino era péssimo para a prática da profissão e para a capacidade de resolver problemas, e com isso este método teve grande aceitação nos meios educacionais (BARROWS, 1996).

Este embasamento difere de abordagens de ensino tradicionais, como o aprendizado passivo. Neste caso, esta abordagem está mais centrada no conceito do aprendizado ativo, colocando o aluno como protagonista da construção de seu conhecimento. Portanto, no PBL, a aprendizagem é dirigida ao incentivo do uso de habilidades de autoaprendizado, onde o aluno tem sua capacidade de procurar e assimilar informações relevantes para a análise de um cenário simulando a realidade, assim, o aluno formulará e priorizará o aprendizado chave, que deve ser de acordo com os objetivos dentro dessa experiência. Também deverá ser entendido as expectativas e motivações dos indivíduos, por isso o feedback também é necessário, para que esse processo seja adequado à real experiência do aprendizado. Isso tudo, utilizando grupos pequenos onde cada indivíduo possui um papel e uma responsabilidade, assim como no dia a dia profissional (BARROWS, 1996, WOODS, 1994; SAVERY, 2006).

O uso de metodologias que envolvam simulações, como o PBL, provem aos estudantes uma variedade de experiência que não são possíveis apenas dentro do contexto acadêmico mais tradicional (YAMPOLSKY, 2016). E neste caso, o aluno poderá construir o pensamento crítico, assim como a consolidação de memórias sólidas sobre os diversos assuntos abordados, a partir da identificação do problema chave e da elaboração de uma possível solução.

2.2 LOW CODE

O termo Low Code surgiu em 2014 após uma publicação da organização norte-americana Forrester (RICHARDSON *et al.*, 2014), na qual anunciava que essas ferramentas traziam mudanças nos métodos, práticas e abordagens no desenvolvimento e entrega de aplicações. Isso se dando pela possibilidade de rápido desenvolvimento de aplicações com o mínimo de esforço na codificação, com o intuito de expressar melhor as lógicas de negócios e minimizar esforços de designer e front-end, assim como integrações e implantação (*deployment*) da aplicação.

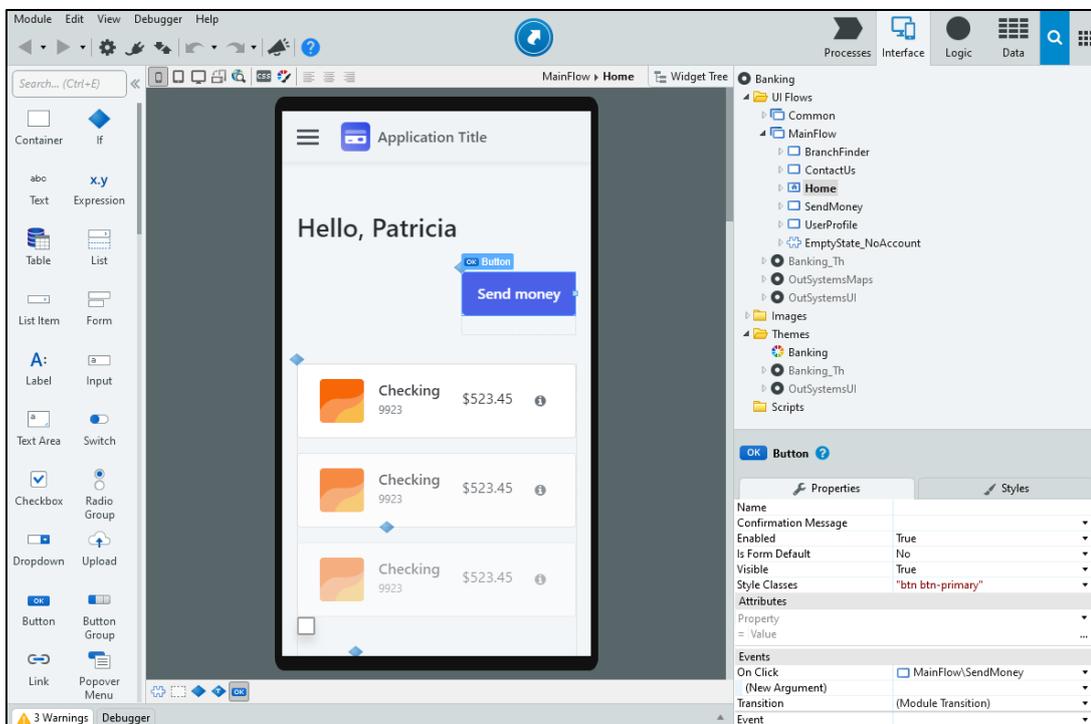
As ferramentas Low Code também podem ser confundidas ou citadas como No Code (ATKINS, 2020). Uma possível causa para isso é o pequeno tempo de vida do termo. Por causa disso, ainda é possível observar uma certa confusão entre os dois conceitos. A diferença entre ambas as ferramentas ainda é um pouco sutil, porém segundo Meijers (s.d.) o Low code é uma ferramenta que requer um pouco mais de conhecimento técnico, com uma boa abertura para customização de código e integrações, já o No Code é uma ferramenta fechada à código e puramente focada na experiência dos usuários, sendo uma plataforma para pessoas não programadoras.

Todos os benefícios dessa ferramenta são proporcionados por diversas características: ferramentas encapsularam a programação visual no seu conceito, trazendo editores visuais para acelerar a entrega de interfaces de usuário (Figura 1), modelos e códigos facilmente interpretáveis e customizáveis com códigos tradicionais e integrações; O uso da metodologia ágil como principal metodologia de construção de software; Aplicação em nuvem (Cloud) (figura 2); Conexão com banco de dados relacionais, como SQL server e ORACLE, com facilidade de operações; Conceitos de segurança embutidos nas ferramentas, como criptografia nas transações entre o banco de dados e papéis; Versionamento e *deployment* contínuo (Figura 3) e; Programação dirigida a modelos e a programação baseada em metadados (RICHARDSON *et al.*, 2014; BAUMGARTEN, 2020; VINCENT *et al.*, 2021).

Para analisar como o uso dessas ferramentas estão inseridas no mercado de software, um questionário feito pela OutSystems (2019) observou que 41% das respostas apontavam que as organizações em que trabalhavam já estavam utilizando uma ferramenta Low Code, e 10% responderam que estavam prestes a começar o uso das ferramentas. Segundo a organização Gartner (VINCENT *et al.*, 2020), é esperado que essa tecnologia alcance a adoção como ferramenta de desenvolvimento em, aproximadamente, 50% das empresas de

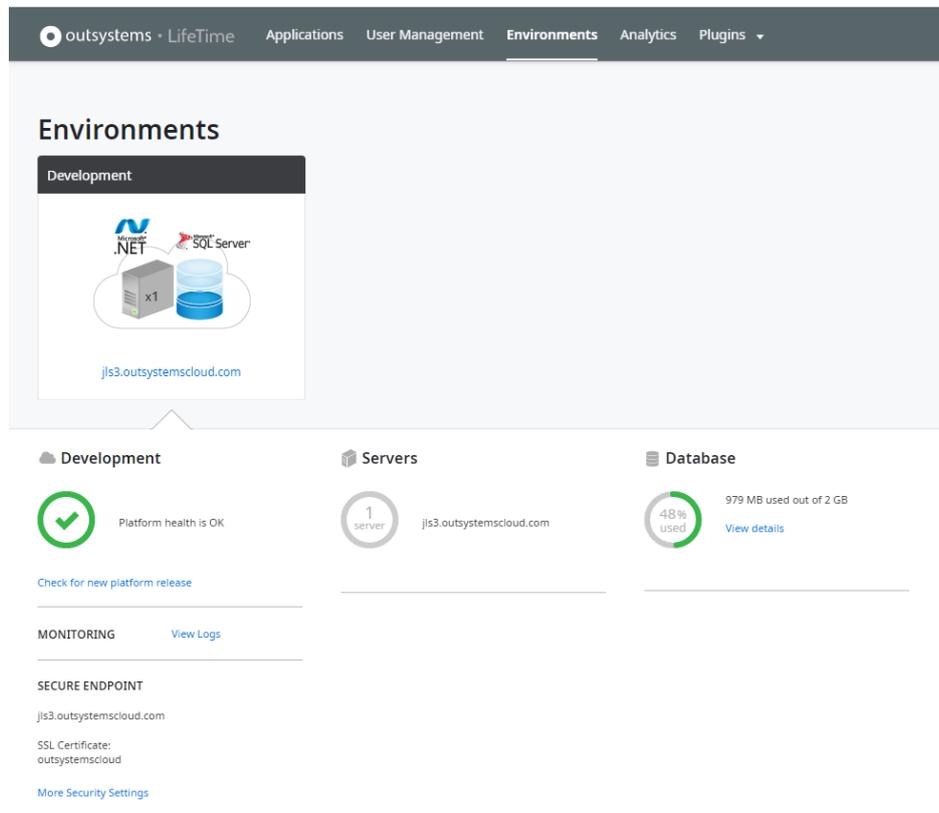
médio e grande porte em 2023. E todo esse crescimento está de acordo com o Low Code Development Platform Market (Marqual IT Solutions Pvt. Ltd, 2020), que reportou que, a tendência do mercado global para esta tecnologia está projetada para crescer de 13,2 bilhões de dólares em 2020 para 46,4 bilhões até 2026, com o crescimento anual de 25% durante o período. Esses números mostram que, de fato, a tecnologia Low Code está bem inserida no mercado, com uma tendência de crescimento muito grande. Também foi identificado pela Gartner (VINCENT *et al.*, 2021) que as organizações que lideram esse nicho são: Salesforce, OutSystems, Microsoft, Mendix, Appian e ServiceNow.

Figura 1 — Editor visual da ferramenta Low Code Outsystems.



Fonte: O autor (2022)

Figura 2 — Gerenciamento de ambientes em nuvem da ferramenta Low Code Outsystems



Fonte: O autor (2022)

Figura 3 — Processo de versionamento e *deployment* facilitado das aplicações da ferramenta Low Code Outsystems

	Development	DEPLOY...	Testing	DEPLOY...	Production
Key Store Plugin	0.1 Feb 21, 05:32 Imelda Nugraha		0.1 May 9, 05:26 Imelda Nugraha		0.1 Jul 29, 14:16 Imelda Nugraha
Lisbon Template	0.3+ 2018-10-22, 01:03 Imelda Nugraha		0.3+ 2018-08-30, 15:59 Imelda Nugraha		0.3 Feb 18, 18:03 Imelda Nugraha
Location	0.1 2018-07-15, 07:21 Imelda Nugraha				
Location Plugin	0.1+ Feb 21, 05:23 Imelda Nugraha				
Logs Management	0.1 Mar 7, 12:20 Imelda Nugraha				

Fonte: O autor (2022)

Como visto anteriormente, o Low Code utiliza o método ágil como seu processo (VINCENT *et al.*, 2021). O método ágil, por coincidência, é o framework mais utilizado para o desenvolvimento de software (SALZA, 2019). Para exemplificar esse fato, uma pesquisa feita no 14º relatório anual State Of Agile (DIGITAL.AI, 2020) identificou que 95% dos participantes apontaram que suas organizações praticavam o método ágil. O aceleração da

entrega de software, o aprimoramento da capacidade de gerenciar mudanças e prioridades, o aumento da produtividade e o melhor alinhamento dos negócios foram as principais razões apontadas para a adoção dos métodos ágeis. Ainda nessa pesquisa, Scrum e suas variantes tiveram 58% das respostas, sendo o método mais utilizado. Na pesquisa feita pela OutSystems (2019), o resultado foi ainda mais promissor, pois o Scrum se mostrou o método mais utilizado nas organizações respondentes com 76%.

O Low Code também é identificado em alguns trabalhos como uma ótima ferramenta para uso em sala de aula, visto que possui extensivos recursos visuais, por ser uma ferramenta com alta produtividade e por ter seu uso facilitado. E por esse fator, também sendo possível observar o uso desta ferramenta em diversos níveis educacionais, do ensino médio ao ensino superior, e com o uso de diversos métodos educacionais associados (LITMAN; MEW e FIELD, 2018).

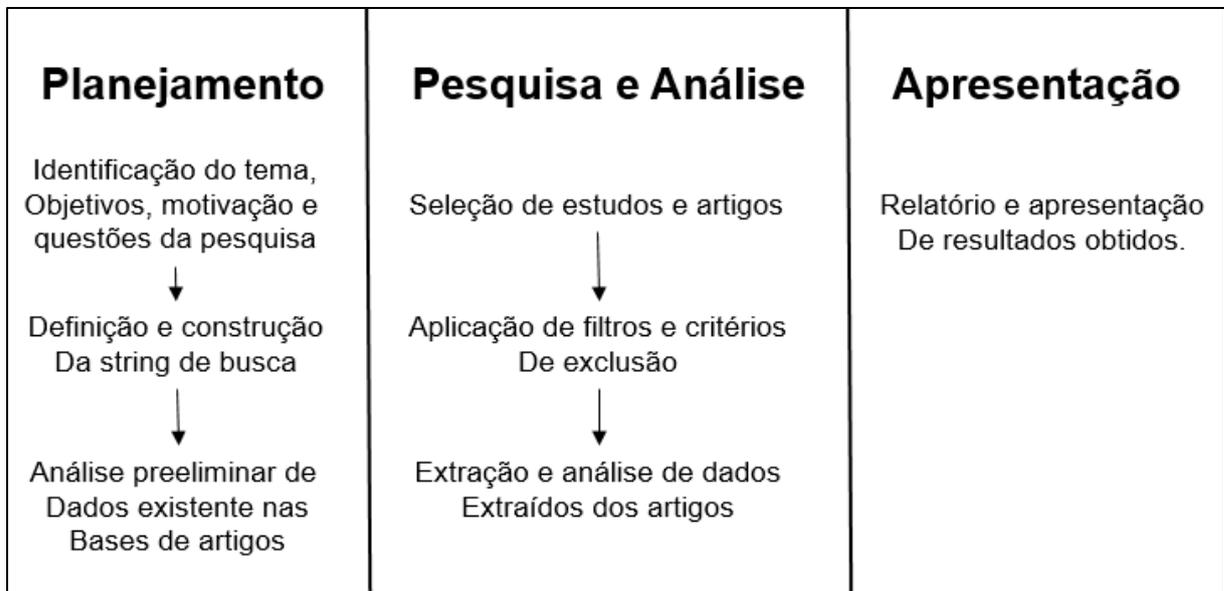
Na literatura, é possível observar o uso do Low Code para fortalecer um conceito chamado de “Cidadão Desenvolvedor” (THACKER, 2021). O Gartner cita que o termo cidadão desenvolvedor considera que colaboradores de uma organização, fora da área de TIC e principalmente na área de negócios, utilizam de tecnologias para gerar recursos em aplicações para uso próprio ou de outros colaboradores (GARTNER, s.d.). Assim, Carvalho (2021) comenta que, a partir dessa teoria, é explorado na educação a promoção da criação de sistemas de informação para suportar negócios com as ferramentas disponíveis na internet, tendo o Low Code como essa possível ferramenta.

3 METODOLOGIA

Em uma análise prévia da literatura, foi observado a ausência de revisões sistemáticas entre Low Code e educação, assim como poucos números de estudos relacionando o uso dessa ferramenta no âmbito educacional. As revisões sistemáticas permitem que seja feito um estudo onde o objetivo é reunir, organizar e analisar estatisticamente artigos relevantes, com um mesmo tema específico e que seja produzido por vários autores. Por consequência disso, foi preciso analisar o tema proposto por esse trabalho em um nível mais amplo, para identificar se existiam estudos relevantes que apontavam o uso do Low Code, exclusivamente, no âmbito educacional. Portanto, para alcançar esse propósito, o presente trabalho utilizou o método de mapeamento sistemático de literatura, descrito por Kitchenham (2007), que possui artifícios necessários para evidenciar o tema e responder a determinadas questões da pesquisa.

O mapeamento sistemático concentra as tarefas necessárias em 3 fases, sendo elas: (1) Planejamento da Pesquisa; (2) Pesquisa e Análise; (3) Apresentação do relatório, conforme observado na Figura 4.

Figura 4 — Etapas do método proposto por Kitchenham (2007).



Fonte: O autor (2022)

Na primeira etapa, o *Planejamento* é feito da seguinte maneira:

1. Identificar a necessidade e/ou motivação para a realização do mapeamento;
2. Especificar perguntas do trabalho;
3. Definir o protocolo do mapeamento:

- a. Qual será o termo de busca;
- b. Quais os critérios para selecionar os estudos;
- c. Como será feita a seleção dos estudos;
- d. Definição dos critérios de inclusão e exclusão dos estudos;
- e. Estratégia para extração dos dados encontrados;
- f. Estratégia para disseminação da revisão;
- g. Cronograma para revisão.

Na próxima etapa, referente às *Pesquisa e Análise*, as atividades são:

1. Realizar pesquisa dos estudos de acordo com a estratégia definida na primeira etapa;
2. Aplicação dos critérios de exclusão;
3. Aplicação dos critérios de qualidade;
4. Extração dos dados;
5. Sintetização dos dados extraídos.

Por último, na etapa relacionada à *Apresentação*, fazer um relatório do mapeamento, as atividades propostas são:

1. Definir como os resultados serão apresentados;
2. Descrever os resultados encontrados;
3. Validação dos resultados descritos.

3.1 PERGUNTAS DA PESQUISA

O mapeamento sistemático deve responder a seguinte questão central: QC - *Como o Low Code está sendo usado no âmbito educacional?* Para isso, quatro questões secundárias foram definidas para avaliar esse questionamento. São elas:

- *Questão 1* - Quais disciplinas e/ou níveis educacionais abordam o Low Code?
- *Questão 2* - Como o Low Code tem sido utilizado?
- *Questão 3* - Por que aprender Low Code (benefícios e vantagens para a formação profissional)?
- *Questão 4* - Quais são os principais desafios, limitações, desvantagens e oportunidades relacionados ao uso de Low Code em educação?

Respondendo estas perguntas, será possível compreender melhor como o Low Code está inserido neste âmbito.

3.2 BUSCA E SELEÇÃO DOS ESTUDOS

Após definir as perguntas da pesquisa, foi iniciada a etapa da coleta de dados dos estudos, em linguagem inglesa ou portuguesa, relacionados ao estudo ou uso de ferramentas Low Code para fins educacionais, com foco no período entre 2016 e 2022. O período foi definido a partir de buscas prévias na literatura científica, evidenciando que a maior parte dos estudos principais se mantiveram nessas datas.

Antes de definir um termo de busca, foram selecionadas dez bases de estudos científicos para a realização das pesquisas. São elas:

- ACM Digital Library (Site: <http://dl.acm.org/>)
- Americas Conference on Information Systems (AMCIS) (Site: <https://aisel.aisnet.org/amcis/>)
- European Conference on Information Systems (ECIS) (Site: <https://aisel.aisnet.org/ecis/>)
- Google Acadêmico (Site: <https://scholar.google.com.br>)
- Hawaii International Conference on System Sciences (HICSS) (Site: <https://aisel.aisnet.org/hicss/>)
- International Conference on Information Systems (ICIS) (Site: <https://aisel.aisnet.org/icis/>)
- IEEE (Site: <https://ieeexplore.ieee.org/>)
- Pacific Asia Conference on Information Systems (PACIS) (Site: <https://aisel.aisnet.org/pacis/>)
- Science Direct (Site: <http://www.sciencedirect.com/>)
- Scopus (Site: <http://www.scopus.com/>)

Após a definição dos repositórios de artigos foi realizada a definição do termo de busca. Para definir quais palavras seriam utilizadas, foram executadas sucessivas tentativas para adaptar o termo de busca na obtenção do melhor resultado, ou seja, maior quantidade de estudos relevantes por busca. Alguns problemas nessa etapa foram encontrados, por se tratar de um mapeamento sistemático o objetivo seria ampliar o termo de busca para abranger o maior número possível de estudos. Porém, ao ampliar o termo com palavras similares, como LowCode, Low-Code, LCNC ou LCAP, os resultados foram piores em algumas bases, com menos estudos encontrados, assim como em outras bases tiveram indiferença quanto ao

número dos estudos. Também foi utilizado a linguagem portuguesa para as buscas, pois foi possível identificar em pesquisas prévias que alguns estudos com Low Code e educação não estavam traduzidos para a língua inglesa. No geral, a palavra Low Code foi suficiente para buscar os estudos, uma vez que a palavra chave foi utilizada em todo o estudo, não apenas no título, resumo e metadados. Como resultado final obtivemos o seguinte termo de busca: ("low code") AND ("education" OR "higher education" OR "school" OR "course" OR "educação" OR "educação superior" OR "escola" OR "curso").

A partir da definição do termo de busca, foi feita a busca automática nas bases de dados selecionadas. Após a aplicação do termo de busca, o primeiro filtro foi aplicado considerando o título e resumo. O segundo filtro levou em consideração a Introdução e Conclusão.

Em ambas as etapas, foram utilizados critérios de exclusão, removendo artigos que não se adequavam ao mapeamento. Esses critérios foram:

- Estudos duplicados;
- Estudos que não apresentam como foco o Low Code ou tecnologias Low Code na educação;
- Revisões sistemáticas;
- Estudos indisponíveis para download;
- Estudos com menos de 3 páginas.

Como critérios de inclusão, foram definidos:

- Estudos completos publicados em conferências e periódicos;
- Estudos na língua inglesa ou portuguesa;
- Estudos dentro do contexto da pesquisa, questão central e secundárias.

Seguindo com a seleção de artigos, foi feita a qualificação dos estudos através de cinco critérios de qualidade definidos no planejamento do mapeamento. Os critérios foram definidos como:

- Contexto claro;
- Metodologia bem definida;
- Aplicação prática;
- Discussão relevantes e consistentes;
- Limitações e ameaças da pesquisa comentadas.

Ao selecionar os estudos, cada critério de qualidade foi avaliado através de uma nota, que variaram de 0 a 1. Os estudos primários com pontuação média de 2,5 foram selecionados para compor o mapeamento. Além disso, as questões de cada estudo foram avaliadas para avaliação do grau de responsividade do artigo para cada questão deste mapeamento:

- 0 – Indica que o estudo não respondeu à pergunta;
- 0,5 - Indica que o estudo respondeu parcialmente à pergunta;
- 1 - Indica que o estudo respondeu totalmente à pergunta.

A partir disso, obteve-se um número de 32 artigos encontrados, portanto, para buscar mais evidências em outros estudos, foi utilizado a estratégia de *snowballing*. Essa estratégia é descrita por Wohlin (2014), por meio da qual é selecionado um conjunto de artigos iniciais, nos quais suas referências serão analisadas de forma recursiva, ou seja, ao selecionar uma referência é analisado se o estudo é viável, após isso, as referências desse estudo também serão analisadas da mesma forma. Neste mapeamento, os 32 artigos foram utilizados como conjunto inicial.

No total, a estratégia de *snowballing* resultou em 12 artigos, que foram submetidos aos mesmos critérios anteriores do mapeamento, resultando em sete artigos selecionados para compor o conjunto de estudos finais para o mapeamento.

No final da extração, um total de 39 estudos foram selecionados, sendo estes os que passaram por todos os filtros aplicados, de exclusão, inclusão e de qualidade. Estes estudos também responderam ao menos uma das questões secundárias definidas na pesquisa. As bases de dados AMCIS, ECIS, HICSS, ICIS, PACIS e Science Direct não tiveram nenhum estudo selecionado após aplicação dos filtros. A Tabela 1 resume os resultados encontrados nas principais etapas do processo de seleção por base de dados. A lista final dos estudos finais se encontra no Apêndice A.

Tabela 1 — Resultados encontrados nas principais etapas do processo de seleção por base de dados.

Fonte	Nº de Estudos	Filtro 1 e Critérios de exclusão	Filtro 2	Critérios de Qualidade
ACM	104	8	3	3
AMCIS	5	2	0	0
ECIS	2	0	0	0
Google Acadêmico	4430	47	22	21
HICSS	1	0	0	0
ICIS	4	0	0	0
IEEE	27	4	1	1
PACIS	1	0	0	0
Science Direct	153	1	0	0
Scopus	214	10	7	7
Snowballing	12	10	7	7
Total	4953	82	40	39

Fonte: O autor (2022)

4 RESULTADOS

4.1 DADOS GERAIS

Dos 39 estudos primários selecionados, 32 estudos foram obtidos a partir de quatro bases: ACM, Scopus, IEEE e Google Acadêmico: ACM, com 3 estudos; Scopus, com 7 estudos; IEEE, com 1 estudo e; Google Acadêmico, com 21 estudos. Os estudos que foram selecionados com o uso do método *snowballing*, foram coletados a partir de pesquisas direta no Google Acadêmico, Research Gate e outros métodos alternativos. Os resultados finais dos estudos podem ser vistos na Tabela 2.

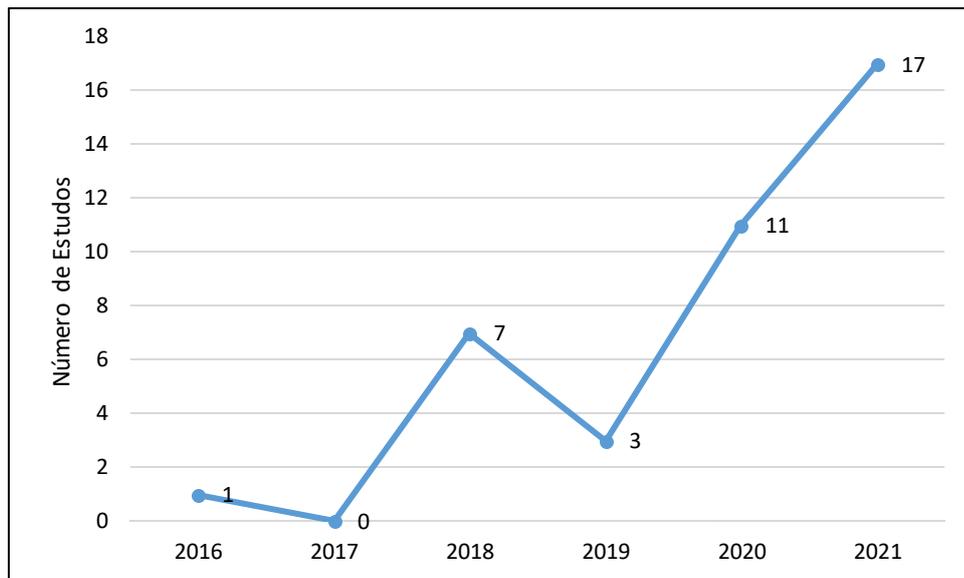
Tabela 2 — Resultados finais dos estudos selecionados para o mapeamento sistemático

Fonte	Nº de Estudos	Frequência (%)
ACM	3	7,7
Google Acadêmico	21	53,8
IEEE	1	2,6
Scopus	7	17,9
Snowballing	7	17,9
Total	39	100

Fonte: O autor (2022)

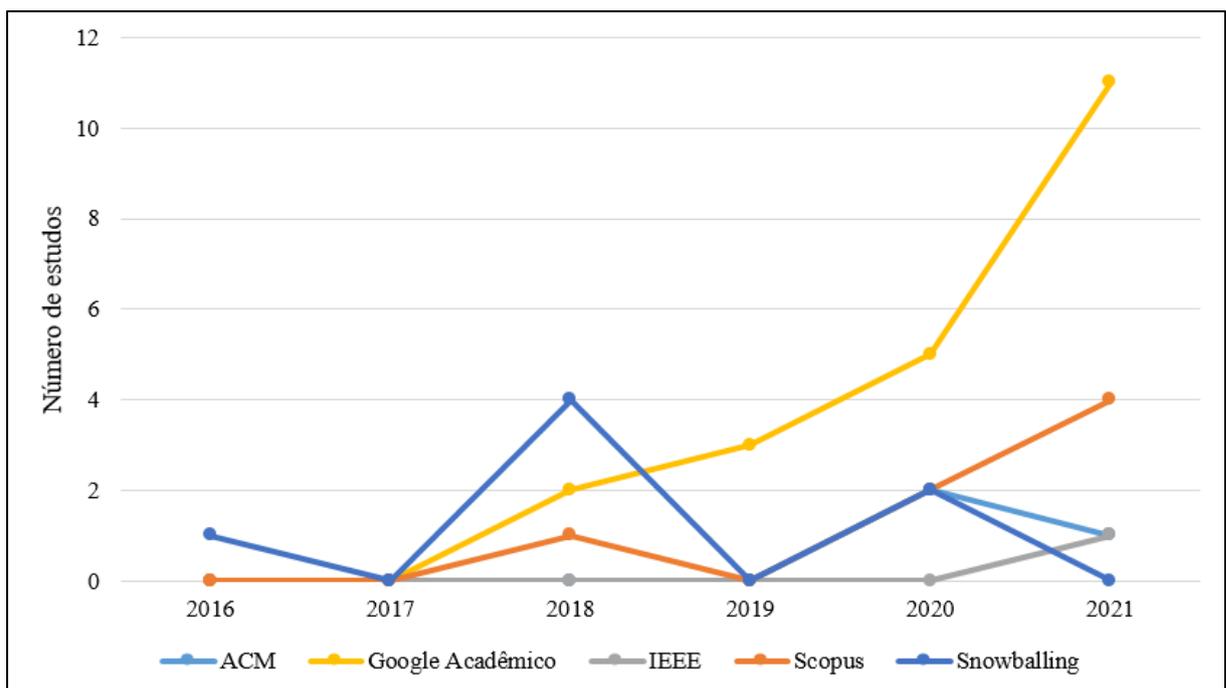
A distribuição de estudos por ano de publicação pode ser vista na Figura 5. Nesse mapeamento, foi identificado que a maior parte dos estudos foram publicados mais recentemente, ou seja, em 2020 e 2021. O que segue a tendência de uso das ferramentas Low Code para os próximos anos no mercado de trabalho (GARTNER, 2021). Na Figura 6, é possível observar os estudos por base de dados e por ano.

Figura 5 — Estudos por ano.



Fonte: O autor (2022)

Figura 6 — Estudos por base de dados e por ano.



Fonte: O autor (2022)

4.2 QUESTÃO 1 - QUAIS DISCIPLINAS E/OU NÍVEIS EDUCACIONAIS ABORDAM O LOW CODE?

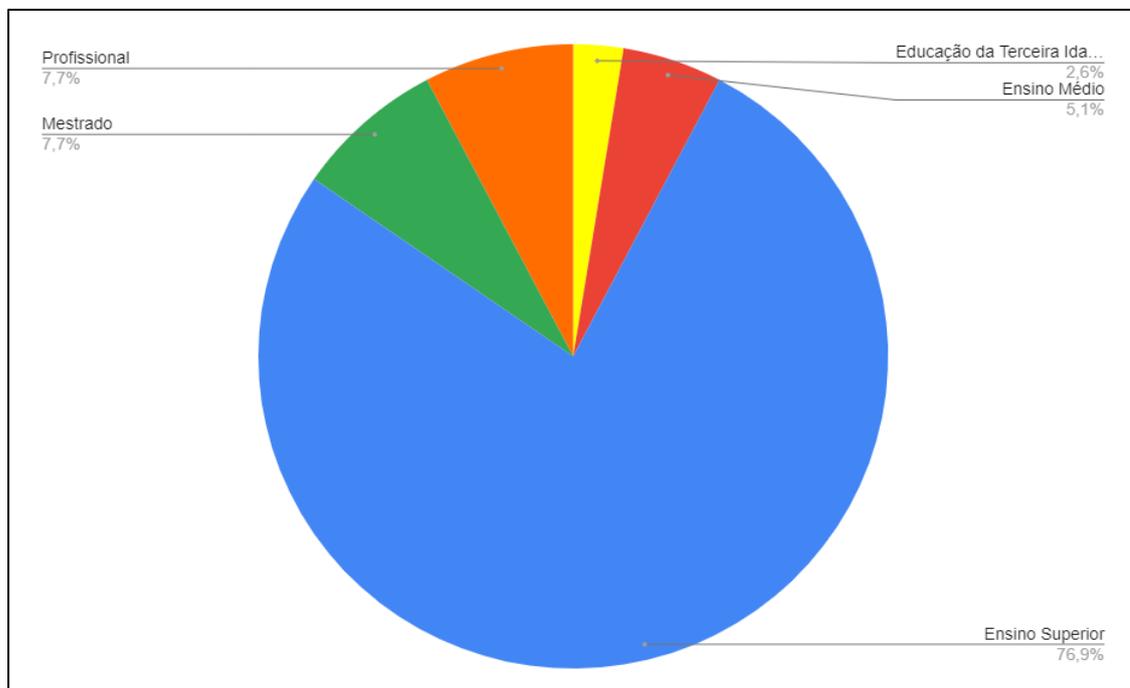
A primeira questão deste estudo se faz necessária para o investigar em quais ambientes e níveis educacionais estão utilizando a tecnologia Low Code. Também possui o propósito de auxiliar nas respostas das outras perguntas, visto que esta pergunta é a base para entender como o uso dessa tecnologia se posiciona em cada ambiente.

Ao analisar o Apêndice B, é possível verificar que o grau de responsividade desta questão é de aproximadamente 96,15%, indicando que foi possível identificar praticamente todos os âmbitos educacionais onde os estudos ocorreram.

4.2.1 Qual o nível educacional?

A partir da análise dos estudos que aconteceram dentro de algum nível educacional foi possível identificar que 76,9% dos estudos estão ligados ao ensino superior, seguido do nível profissional 7,7%, mestrado 7,7%, ensino médio 5,1% e educação na terceira idade 2,6% (Figura 7; Tabela 3).

Figura 7 — Distribuição dos níveis educacionais do mapeamento.



Fonte: O autor (2022)

Tabela 3 — Níveis educacionais identificados nos estudos do mapeamento.

Nível educacional	Estudos
Educação da Terceira Idade	EP13
Ensino Superior	EP1, EP2, EP3, EP4, EP5, EP6, EP7, EP9, EP11, EP12, EP14, EP15, EP17, EP18, EP19, EP20, EP21, EP22, EP25, EP26, EP27, EP28, EP30, EP31, EP32, EP33, EP34, EP35, EP36, EP38
Ensino Médio	EP8, EP37
Mestrado	EP5, EP15, EP33
Profissional	EP10, EP16, EP39

Fonte: O autor (2022)

Essa grande proporção de estudos na educação superior já era esperada, visto que é nesse nível educacional onde reside grande parte de cursos superiores que possuem foco ou utilizam da programação para os fins de aprendizado, resolução de problemas, preparo para o mercado de trabalho e para contribuições sociais, como projetos de extensão. Porém, ainda foi possível identificar estudos feitos concomitantemente com o Ensino Superior e o mestrado [EP5; EP15; EP33]. Um ponto interessante é que o uso do Low Code também foi identificado no âmbito profissional, ensino médio e na Educação da terceira idade. Algumas evidências da informação coletada:

“This paper discusses the experience of using Mendix, a Low Code platform, to support the goals and objectives of an undergraduate project management course in information systems curriculum for students who are continuing studies in a bachelor’s program.”. [EP3]

“This project attempts to understand the technological self-conception of business students with regards to being a technological developer versus simply being a technology user in preparation for incorporating low-code/no-code tools and workflow automation activities into the curriculum”. [EP18]

“An exploratory, six-week custom developed course in information systems was created, entitled Agile Low-Code Development”. [EP26]

4.2.2 Quais cursos, disciplinas e áreas do conhecimento?

Foi possível identificar o uso do Low Code em vários cursos e/ou disciplinas (Tabela 4). De acordo com a Figura 8, é possível observar a concentração de áreas do conhecimento dos estudos mapeados.

Interessante notar que, apesar da área de TIC manter a maior concentração dos estudos (58,9%), há uma variedade de áreas onde o Low Code se encaixa, como na educação geral (12,5%), que engloba a educação global, curso online, curso rápido, ensino médio, programa interdisciplinar e seminário. Também foi possível observar estudos nas áreas de design, administração, comércio, agricultura e no treino de profissionais desempregados.

Ainda vale ressaltar que o uso do Low Code também está inserido na extensão universitária e na pesquisa, sendo esta uma ferramenta potencial para atuar no princípio de indissociabilidade ou “tripé universitário”, estabelecido no artigo 207 da constituição Brasileira. Este princípio constitui os pilares da universidade no Brasil, ou seja, a missão e dever dessa instituição, que é formado pelo ensino, pesquisa e extensão (BRASIL, 1990).

Tabela 4 — Relação entre curso e/ou disciplina e estudos. (Continua)

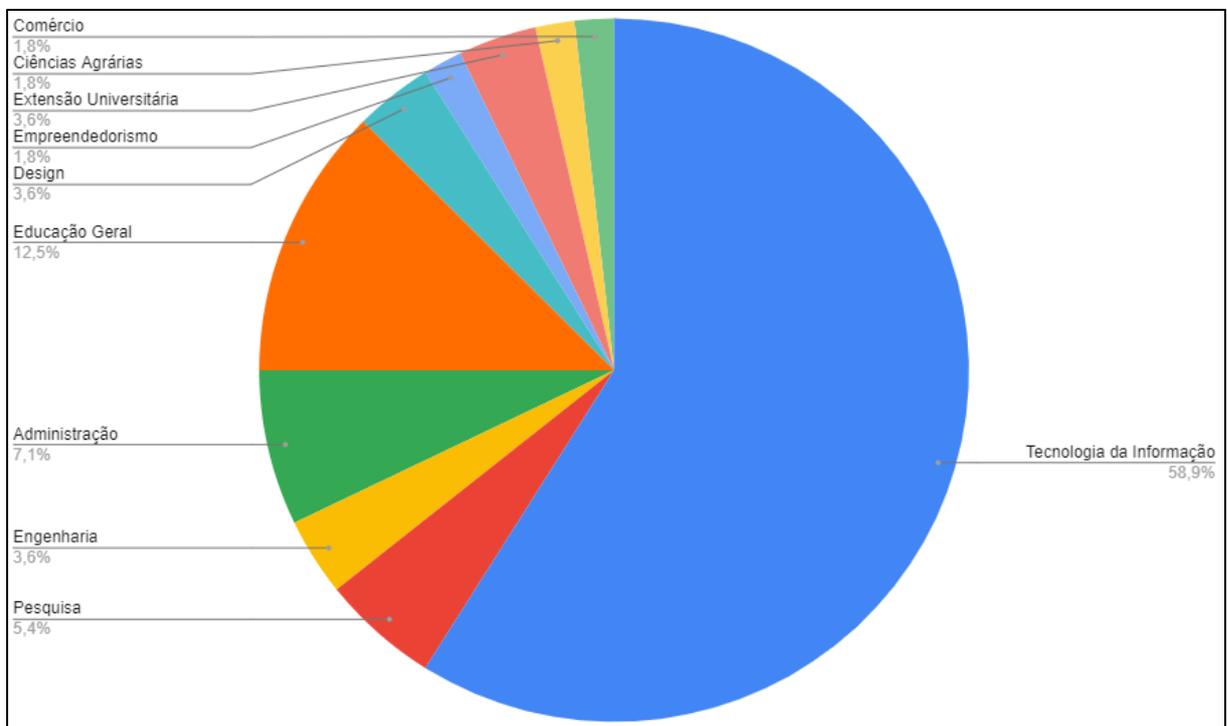
Curso/Disciplina	Estudos
Administração	EP4, EP18, EP33, EP36
Agricultura	EP24
Ambiente Organizacional	EP39
Análise de Sistemas	EP9, EP17
Banco de Dados	EP22
Ciência da computação	EP2, EP9, EP21, EP38
Comércio	EP33
Computação Geral (Bacharelado, Licenciatura)	EP1, EP20, EP33, EP35
Curso Online	EP34
Curso Rápido	EP10
Desenvolvimento de Software	EP11, EP25, EP26
Design	EP17, EP30
Educação Global	EP29
Empreendedorismo	EP23
Engenharia	EP4, EP14
Engenharia da Computação	EP5, EP27
Engenharia de Software	EP7, EP12, EP21, EP31
Engenharia Orientada a modelos	EP15
Ensino Médio	EP8
Extensão Universitária	EP23, EP37
Gerenciamento de Projetos	EP3
Pesquisa	EP2, EP13, EP28

Tabela 4 — Relação entre curso e/ou disciplina e estudos.

Curso/Disciplina	Estudos
Programa Interdisciplinar	EP32
Qualidade de Software	EP9
Seminário	EP15
Sistema de Informação	EP1, EP3, EP5, EP6, EP20
Tecnologia da Informação e Comunicação	EP1
Transformação Digital	EP19
Treino de Profissionais Desempregados	EP10, EP16

Fonte: O autor (2022)

Figura 8 — Distribuição das áreas do conhecimento.



Fonte: O autor (2022)

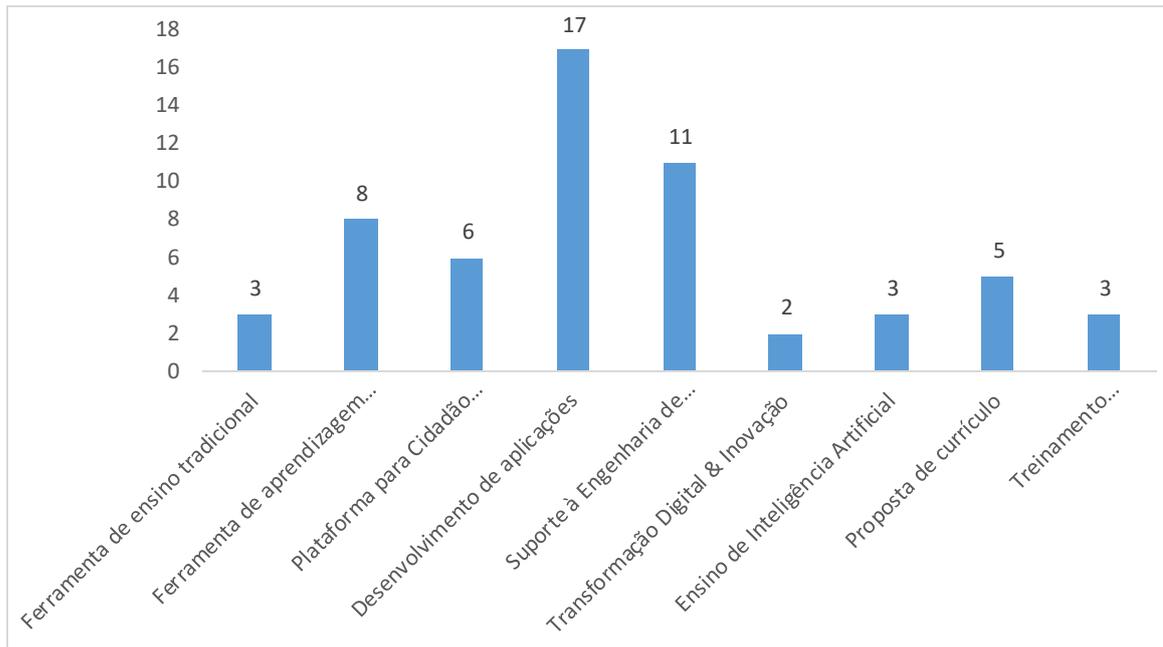
4.3 QUESTÃO 2 - COMO O LOW CODE TEM SIDO UTILIZADO?

A segunda questão tem o intuito de entender como o Low Code tem sido utilizado nos diferentes ambientes educacionais para que seja ampliado a visão de como este recurso é benéfico e, também, para edificar uma resposta ressonante para a contribuição desta tecnologia de desenvolvimento na educação.

Com uma porcentagem de 80,77% de responsividade, dentro do universo de dados adquirido através desse mapeamento (Apêndice B), essa pergunta foi bem respondida. É importante notar que a maior proporção de uso do Low Code se manteve coerente com sua principal finalidade, o desenvolvimento de aplicações e, também, no Suporte à Engenharia de Software (Simulação, TDD, BDD, Prototipagem, Métodos Ágeis) (Figura 9; Tabela 5). Ferramenta de aprendizagem ativa o conceito de cidadão desenvolvedor foram identificados com boas proporções de estudos. Também foi possível identificar metodologias relativas à engenharia de software, como Desenvolvimento guiado por testes (TDD) e Desenvolvimento orientado por comportamento (BDD) [EP7]. O estudo [EP26] utilizou tanto os métodos ágeis como o desenvolvimento de aplicações para desenvolver soluções, como destacado a seguir:

“The course enabled participants to develop applications using the Mendix Low Code Platform and the Agile project methodology. The course achieved three primary goals: (1) to teach the basic concepts of Agile from a holistic perspective; (2) to develop the knowledge and ability to use the Mendix Low-Code Platform; and 3) to develop an application using Agile and Mendix in a group project.”

Figura 9 — Gráfico mostrando o número de estudos identificados para cada uso da ferramenta.



Fonte: O autor (2022)

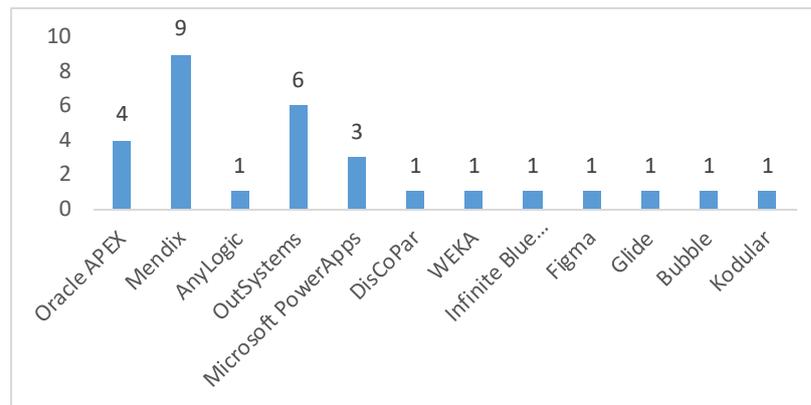
Tabela 5 — Estudos que responderam como as ferramentas Low Code são utilizadas.

Usos	Estudos
Ferramenta de ensino tradicional	EP30, EP37, EP39
Ferramenta de aprendizagem ativa (PBL, PrBL, outras)	EP7, EP8, EP9, EP12, EP16, EP17, EP21, EP35
Ferramenta para Cidadão Desenvolvedor	EP18, EP19, EP23, EP29, EP34, EP36
Desenvolvimento de aplicações	EP1, EP3, EP5, EP6, EP8, EP10, EP14, EP15, EP16, EP20, EP24, EP25, EP26, EP33, EP34, EP35, EP38 (jogos)
Suporte à Engenharia de Software (Simulação, TDD, BDD, Prototipagem, Métodos Ágeis)	EP4, EP3, EP7, EP11, EP13, EP20, EP21, EP25, EP26, EP33
Transformação Digital & Inovação	EP19, EP39
Ensino de Inteligência Artificial	EP27, EP28, EP29
Proposta de currículo	EP2, EP17, EP18, EP31, EP32
Treinamento profissional/Extensão	EP10, EP16, EP22

Fonte: O autor (2022)

Também foi possível mapear as ferramentas Low Code utilizadas (Figura 10). A ferramenta Mendix foi a mais utilizada, seguida de OutSystems, Oracle APEX, e Microsoft PowerApps.

Figura 10 — Ferramentas Low Code utilizadas nos estudos.



Fonte: O autor (2022)

A Tabela 6 relaciona os usos do Low Code com os níveis educacionais, mapeando quais estudos usaram o Low Code nos diferentes níveis educacionais. Aqui se consegue rastrear bem como essa tecnologia se enquadra em cada nível de ensino.

Como esperado, diante da suposta facilidade do uso das ferramentas Low Code, foi possível observar, no ambiente educacional, o uso desta tecnologia em níveis além do ensino superior e do mestrado. E a partir desse pressuposto, ao analisar os estudos mapeados, foi possível observar evidências de uso na educação da terceira idade, trazendo o Low Code como ferramenta para os pesquisadores que desenvolveram protótipos para a utilização na educação tecnológica de mulheres acima dos 80 anos [EP13].

No Ensino médio, o Low Code foi utilizado para introduzir o ensino do desenvolvimento de aplicativos mobile em programas de ensino de programação para estudantes do 9º ano do ensino fundamental ao 2º ano do ensino médio no Brasil [EP37]. Em outro estudo, por meio da ferramenta OutSystems, o Low Code foi utilizado no processo de aprendizagem ativa, com o intuito de ensinar como funciona o desenvolvimento de aplicativos [EP8].

No nível profissional, essa tecnologia foi empregada com dois objetivos, são eles: estimular o desenvolvimento de inovação [EP39] e oferecer treinamento profissional para pessoas desempregadas, sejam elas profissionais da área de TIC [EP10] ou profissionais da área de ciência, tecnologia, engenharia artes e matemática (STEAM) [EP16].

Tabela 6 — Relação dos estudos na utilização do Low Code nos diferentes níveis educacionais.

	Educação					
	da Terceira Idade	Ensino Médio	Ensino Superior	Mestrado	Profissional	Outros
Ferramenta de ensino tradicional	-	EP37	EP30	-	EP39	-
Ferramenta de aprendizagem ativa (PBL, PrBL, outras)	-	EP8	EP7, EP9, EP12, EP17, EP21, EP35	-	EP16	-
Ferramenta para Cidadão Desenvolvedor	-	-	EP18, EP19, EP34, EP36	-	-	EP23, EP29
Desenvolvimento de aplicações	-	EP8	EP1, EP3, EP5, EP6, EP14, EP15, EP20, EP25, EP26, EP33, EP34, EP35, EP38 (jogos)	EP15, EP33	EP10, EP16	EP24
Suporte à Engenharia de Software (Simulação, TDD, BDD, Prototipagem, Métodos Ágeis)	EP13	-	EP3, EP4, EP7, EP11, EP20, EP21, EP25, EP26, EP33	EP33	-	-
Transformação Digital & Inovação	-	-	EP19	-	EP39	
Ensino de Inteligência Artificial	-	-	EP27, EP28	-	-	EP29
Proposta de currículo	-	-	EP2, EP17, EP18, EP31, EP32	-	-	-
Treinamento profissional/Extensão	-	-	EP22	-	EP10, EP16	-

Fonte: O autor (2022)

Já no ensino superior, foi possível observar maior parte dos usos foram para o desenvolvimento de aplicações e o uso como ferramenta de aprendizagem ativa, que se baseia em métodos como PBL, PrBL, métodos ágeis e outras. E nesses casos, a grande parte dos estudos foram relacionados às áreas relacionadas com TIC, como engenharia de software, sistema de informação, ciência da computação, entre outros. Porém, foi possível verificar que outras áreas também utilizaram dessas metodologias, tal como o estudo no curso de engenharia industrial, que estava envolvido com o desenvolvimento de aplicações [EP14]. No Estudo [EP17], foi empregado o método PrBL no curso de designer. Em [EP34] o estudo descreve um curso online oferecido para o nível superior, que empregou ferramentas Low Code no desenvolvimento de software.

Alguns estudos utilizaram o Low Code para criar modelos de inteligência artificial, utilizando uma biblioteca Low Code para Python [EP28]. Um outro estudo utilizou a ferramenta WEKA para criar modelos para identificar fatores de aprendizado em estudantes [EP27]. Vale salientar que apenas um estudo foi utilizado para a criação de um chatbot com métodos Low Code [EP15].

Em relação ao conceito de cidadão desenvolvedor no ensino superior, quatro estudos foram identificados relacionando este conceito, sendo dois deles fora da área de TIC, no curso de administração [EP18; EP36]. Esses estudos mostram a importância que a ferramenta traz para os profissionais dessa área.

Um outro ponto importante é observado quando o Low Code está encapsulado em alguma proposta de currículo. Como no [EP2] que traz a ferramenta Mendix para solucionar “buracos” presentes na educação de programação, o estudo de implementação dessa mesma ferramenta em um curso de designer e análise de sistema [EP17] e a associação desse conceito com o do cidadão desenvolvedor no curso de administração [EP18]. E mais dois relatos na área de TIC como o uso do Low Code para consolidar educação em Engenharia de Software [EP31] e na melhoria da infraestrutura e experiência educacional dos desenvolvedores estudantes [EP21].

Outros usos curiosos no ensino superior foram:

- Criação de jogos através de uma ferramenta Low Code para trabalhar com grafos [EP38];
- Criação de simulações em cursos que não utilizam simulações [EP4];
- Transformação digital de ferramentas de um Campus [EP19];
- Criação de tutoriais em outras línguas para alunos [EP22].

Já para o mestrado apenas dois estudos utilizaram Low Code [EP15; EP33], ambos também são relacionados com o ensino superior. Sendo os dois relacionados ao desenvolvimento de aplicações e apenas o [EP33] com ênfase em métodos ágeis.

No contexto da extensão, foram identificados três estudos: O estudo [EP23], que utiliza o Low Code como ferramenta para familiarização da programação em um curso de extensão, que fomentou o empreendedorismo por meio do conceito de cidadão desenvolvedor; o estudo [EP29], que estudou a criação de modelos de inteligência artificial aplicado ao bem-estar social, para fácil utilização de qualquer pessoa que se enquadre no conceito de cidadão desenvolvedor; e o estudo [EP24], que utilizou a ferramenta DiscoPar para utilizar na agricultura, para auxiliar fazendeiros a criar suas próprias aplicações com foco no seu dia a dia.

4.4 QUESTÃO 3 - POR QUE APRENDER LOW CODE (BENEFÍCIOS E VANTAGENS PARA A FORMAÇÃO PROFISSIONAL)?

Como mostrado na Tabela 7, foi detectado um ótimo número de benefícios e vantagens do aprendizado do Low Code para a formação profissional. No total, 33 estudos apresentaram alguma resposta para essa questão, restando apenas 6 sem resposta [EP13; EP20; EP22; EP27; EP28; EP35].

Tabela 7 — Benefícios e vantagens do aprendizado do Low Code para a formação profissional.

Benefícios e vantagens	Estudos
Produtividade no desenvolvimento de software (alto nível de abstração, implantação e modularização facilitada, e automação e otimização de processos)	EP1, EP5, EP6, EP8, EP10, EP11, EP12, EP14, EP15, EP16, EP17, EP19, EP21, EP25, EP30, EP31, EP32, EP36, EP38
Facilidade em alcançar objetivos educacionais relacionados a área de TIC e Aprendizado da ferramenta Low Code	EP1, EP3, EP10, EP11, EP12, EP14, EP16, EP17, EP23, EP24, EP25, EP26, EP29, EP36
Construção e melhoria de habilidades (colaboração, solução de problemas, construção novas habilidades)	EP6, EP8, EP10, EP11, EP12, EP16, EP21, EP25, EP30, EP32
Competitividade e treinamento no mercado de trabalho	EP2, EP3, EP4, EP5, EP6, EP7, EP8, EP10, EP14, EP21, EP33, EP34, EP37, EP39
Desenvolvimento de soluções criativas	EP18, EP19, EP39
Suporte ao aprendizado em banco de dados (modelagem e scripts)	EP3, EP10, EP12
Experiência de usuário	EP11
Segurança	EP11

Fonte: O autor (2022)

Entre os benefícios e as vantagens encontradas foi possível destacar características coerentes com a descrição do relatório do Gartner sobre Low Code (VINCENT et al., 2021). A principal delas é a produtividade no desenvolvimento de software, que está ligada a diversos conceitos identificados nos estudos selecionados. Alguns desses estudos citam que o alto nível de abstração das ferramentas Low Code se mostrou benéfica na produtividade de software, que por consequência, possibilitou ao aluno a vantagem de aplicar facilmente conceitos teóricos do desenvolvimento de software [EP1; EP6; EP8; EP11; EP15; EP21; EP25; EP30; EP38]. Já outros estudos também citaram: a facilidade em fazer *deployments* dentro da ferramenta [EP10; EP11; EP12]; a facilidade em modularização, o que facilita entendimento de arquitetura de software [EP15]; o desenvolvimento e a manutenção da aplicação em nuvem, que no geral é fornecido pelo próprio fornecedor da ferramenta, como

exemplo a ferramenta OutSystems [EP5; EP8; EP14]; além de exigir esforço mínimo com codificação, o que pode tornar o processo de desenvolvimento menos burocrático [EP19].

Sobre o alto nível de abstração, como dito anteriormente, foi possível observar que alguns estudos citaram a vantagem de aplicar facilmente conceitos teóricos do desenvolvimento de software. O que mostrou uma influência benéfica para os próximos benefícios e vantagens que serão apresentadas a seguir. As evidências são:

“Additionally, the high level of abstraction makes these tools excellent for demonstrating and enhancing student acquisition of traditional management information systems (MIS) concepts such as use cases, domain modeling, business processes and Agile methodologies.”.
[EP6]

“This special topic course in Agile Low-Code development is an exciting endeavor, since the expectation is that the higher level of abstraction allows more students to develop competencies in the course material.”. [EP11]

“All in all, Xatkit proposes a higher-abstraction solution to the chatbot domain that combines the benefit of platform independent chatbot definition, including non-trivial chatbot actions and side effects, together with an easy deployment.”. [EP15]

Alguns estudos reportaram características que indicaram que o uso das ferramentas Low Code trouxe vantagens associadas à facilidade em alcançar objetivos educacionais relacionados a área de TIC e no aprendizado da própria ferramenta Low Code. Uma característica dessa vantagem identificada foi o foco no aprendizado dos métodos ágeis, uma vez que a facilidade no desenvolvimento permitiu com que os alunos pudessem focar nas entregas, artefatos e cerimônias relativas ao gerenciamento de projeto [EP25; EP26]. Os estudos citados também afirmaram que mais alunos participaram da fase de codificação devido à característica de não necessitar de grande conhecimento específico em linguagens de programação [EP1; EP10; EP12; EP14; EP16; EP17; EP23; EP24; EP26; EP29; EP36].

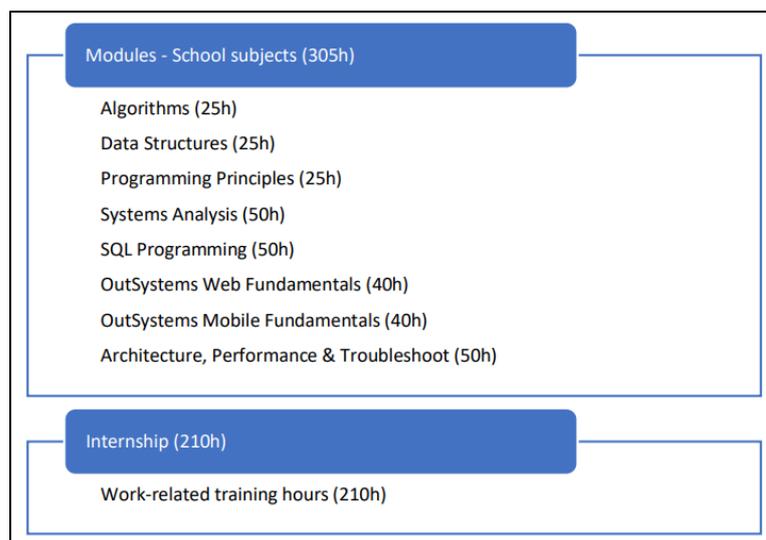
O Low Code também foi protagonista da construção e melhoria de habilidades, como o uso da ferramenta de forma colaborativa [EP30; EP32], na solução de problemas propostos nas aulas [EP8; EP12], na construção novas habilidades na área de TIC [EP6; EP8; EP10; EP16; EP21; EP25] e no treinamento de habilidades [EP10].

O estudo [EP10] é um bom exemplo para responder à terceira pergunta deste mapeamento. Neste trabalho, foi utilizado a ferramenta OutSystems no retreinamento de profissionais para a área de TIC. Este treinamento foi composto por um curso de 305 horas

mais um estágio de 210 horas na área, de acordo com o plano de curso apresentado na Figura 11. Segundo o autor, essa abordagem foi utilizada porque as ferramentas de desenvolvimento Low Code fornecem alguns benefícios como editores visuais, que auxiliam na programação, possibilitando que os usuários não necessitem de grande conhecimento específico em linguagens de programação, ou seja, anos de experiência nessas linguagens para a criação de aplicações complexas. Esse aspecto torna essa ferramenta perfeita para ser rapidamente aprendida por profissionais com habilidades técnicas da área de STEAM, sendo assim, um facilitador para a entrada desses profissionais em áreas com boa empregabilidade, como destaca o estudo [EP10]:

“For the design of a course for reskill professionals to ICT, we consider that the OutSystems low-code development platform fits perfectly due to the following features: • High-productivity visual development to deploy and manage applications; • The ability to deploy Applications and update them with one click in a cloud environment; • Handling database scripts and deployment processes reducing the skills to develop a real application; • Ultra -fast visual modelling mobile applications with offline data synchronization, native device access, and on-device business logic.” [EP10]

Figura 11 — Plano de curso para retreino de profissionais na área de TIC.



Fonte: [EP10]

Também foi importante perceber que a competitividade e treinamento no mercado de trabalho também teve um destaque neste mapeamento. Isso se deu por consequência de uso das ferramentas para simular o ambiente de trabalho e para o treinamento de profissionais, o que torna os alunos competitivos no mercado de trabalho, visto que essas ferramentas já possuem grande parte do mercado de software (VINCENT et al., 2021). E estes aspectos também são importantes para o ganho de experiência de um desenvolvedor de software, pois o aluno pode aprender, de forma integrada, tudo o que um profissional de TIC utiliza em seu dia a dia de trabalho, como enfatiza os estudos:

“As these tools gain momentum in the marketplace, a need exists to give students the competencies to use these tools to keep them relevant and competitive in the job market.”. [EP6]

“Another important initiative to enable students to get in touch with practice in software engineering is a one-day visit to the premises of another software house company. [...]

[...]During this journey, students were able to have closer contact with some Scrum activities (namely Daily Scrum, Sprint, Sprint Execution) and contact with some SCRUM Roles (Scrum Master, Development Team).”. [EP7]

“[...] Using the Mendix platform in university education provides opportunities to embed employability [...]”. [EP33]

“O ensino da programação dentro do Ensino Básico traz à classe mais jovem a vasta gama de possibilidades de criação e desenvolvimento de algoritmos, bem como todas as possibilidades de ingresso no mercado de trabalho através deste ramo.”. [EP37]

Ainda sobre o mercado de trabalho, também foi possível identificar a utilização dessas ferramentas no ensino médio, demonstrando que são ótimas para o desenvolvimento do aprendizado de lógica de programação e para a aquisição de novas habilidades, o que pode auxiliar e abrir um leque para o mercado de trabalho para estes alunos [EP8; EP37].

Também foi identificado um estudo citando vantagens e benefícios na área de experiência de usuário, que foram trazidas a partir da ferramenta Low Code e também na segurança [EP11].

4.5 QUESTÃO 4 - QUAIS SÃO OS PRINCIPAIS DESAFIOS, LIMITAÇÕES, DESVANTAGENS E OPORTUNIDADES RELACIONADOS AO USO DE LOW CODE EM EDUCAÇÃO?

A quarta questão é fundamental para o fechamento do mapeamento, instigando questionamentos para futuras investigações e possíveis trabalhos para sanar os desafios trazidos pelo uso dessa tecnologia no âmbito educacional. Apesar de que a responsividade dessa questão tenha sido de apenas 64% (Apêndice B), a lista de oportunidades e desafios, limitações, desvantagens e desafio mostrou ser coerente com as respostas já discutidas pelas outras questões e pelos benefícios descritos no relatório do Gartner (VINCENT *et al.*, 2021).

4.5.1 Quais são os desafios, limitações e desvantagens?

Os desafios, limitações e desvantagens do Low Code no âmbito educacional foram listados neste trabalho através da Tabela 8. Apenas 12 tópicos foram identificados, restritos a oito artigos, que citaram os desafios, o que pode indicar um possível viés dos estudos analisados [EP11, EP12, EP19, EP20, EP21, EP30, EP32, EP39].

Tabela 8 — Desafios, limitações e desvantagens relacionados ao uso do Low Code no âmbito educacional.

Desafios, limitações e desvantagens	Estudos
Backlog Comprometido	EP21
Declínio no Engajamento	EP39
Desconhecimento da Ferramenta	EP12
Difícil de Dominar	EP20
Dificuldade em Aprender	EP11
Engajamento Total	EP21
Grande Volume de Uso de Dados	EP30
Licença	EP32
Limitações do Low Code	EP19
Pouca Qualidade na Inovação	EP39
Poucas Funcionalidades Pre-construídas	EP30
Sem Portabilidade	EP32

Fonte: O autor (2022)

O comprometimento do backlog e o engajamento total, foi trazido pelo estudo [EP21]. O problema observado não é derivado de uma ferramenta ou metodologia com falhas, mas pelo fato de que o desenvolvimento é rápido e facilitado. Dessa forma, se faz necessário um grande engajamento do *product owner* (o líder de projeto na metodologia Scrum), com grande presença nas atividades mais complexas, mantendo um alto nível de engajamento entre o time de desenvolvimento e os analistas de negócio. Esse ponto traz a visão de que as organizações devem se adequar ao modelo de negócio trazido pelo Low Code, necessitando treinar e avaliar riscos de comprometimento no backlog, tanto para o alto risco com a sincronização do time, como também para as estimativas do backlog do projeto. Na educação, o professor deve tomar cuidado ao assumir ou delegar o papel de *product owner*, pois essa desvantagem pode ser um risco associado ao tempo do desenvolvimento durante a disciplina, assim como trazer uma alta carga de trabalho para o professor.

Um outro ponto observado foi a possibilidade de que a ferramenta pode causar problemas ao processo de inovação, visto que com o passar do tempo pode causar declínio no engajamento, porém esta visão não fica muito clara no estudo [EP 39], que traz maneiras de flexibilizar e inovar com o uso de ferramentas Low Code.

No artigo [EP12] é citado o desconhecimento da ferramenta OutSystems pelos alunos, pois nunca tinham ouvido falar sobre ela. Diante disso, é justificado a importância do trabalho feito, que utilizou o Low Code para atingir escalabilidade numa disciplina baseada em projeto através do método PrBL. Porém isso pode ser um problema para o processo de entendimento do que é o Low Code, de modo que o professor seja o responsável pela apresentação da ferramenta.

O estudo [EP20] cita uma dificuldade em dominar o Mendix. Apesar de ter sua característica como Low Code, alguns alunos, até mesmo os que possuíam algum background, acharam difícil de dominar a tecnologia [EP20]. Porém, não é possível extrapolar essa visão para todas as ferramentas Low Code, visto que este estudo utilizou a ferramenta Mendix.

Apesar de reconhecer que o Mendix é útil para a entrega de pequenos projetos, o [EP11] cita a dificuldade que os alunos tiveram em aprender Mendix e metodologia ágil em uma disciplina com regime especial de seis semanas. No estudo em questão o curso possuía o formato de 16 semanas. Como o estudo foi inserido com um formato reduzido, o impacto negativo pode ter sido aumentado:

“Challenges faced by the students on Mendix technical issues and questions caused the majority of uneasiness. The six-week duration of the course was insufficient for some students to develop the level of competence desired for app development, and feedback suggested that learning both Agile and Mendix competencies in six weeks proved to be difficult.”. [EP11]

O estudo [EP 30] traz a visão do grande volume de uso de dados, ocasionado pela aplicação em nuvem. O que deve ser visto como uma desvantagem, impossibilitando a codificação off-line e o seu uso em ambientes educacionais que não possuam uma boa conexão para o uso das plataformas. Este estudo também citou que para o uso em sala de aula a ferramenta Low Code possui poucas funcionalidades pré-construídas. Este problema pode ser um impeditivo para alunos com pouca expertise em codificação.

A limitação relacionada à licença foi relatada no estudo [EP32]. O autor cita que as ferramentas Low Code não possuem um modelo de precificação universal, ficando a cargo do próprio fornecedor precificar. Isto pode gerar dificuldades ao produzir comparações dos custos de hospedagem e licença. As taxas desse tipo de ferramenta podem variar significativamente e isso depende de diversos fatores. Alguns modelos levam em consideração a forma de hospedagem para basear sua precificação, também podem considerar cobranças por usuários ativos e aplicações construídas. Este estudo também fala da falta de portabilidade de código, isto é, não existem mecanismos de conversão de código para outros ambientes ou tecnologias, portanto os aplicativos precisam ser construídos do zero.

O estudo [EP19] afirma que para este tipo de ferramenta de desenvolvimento sempre haverá algum problema nos requisitos do projeto que não poderá ser resolvido. Fazendo com que esse desafio seja solucionável com abordagens clássicas de desenvolvimento de software. Porém algumas ferramentas já conseguem lidar com esses desafios através de extensões de código e integrações, como Outsystems que possui extensões de códigos em .NET para adicionar camadas de códigos customizados nas aplicações (OUTSYSTEMS, 2022).

Já o estudo [EP39] fala que mesmo o Low Code mostrando potencial na redução da barreira para o desenvolvimento de aplicativos inovadores, também pode trazer novos desafios que podem ocasionar resultados de inovação de baixa qualidade nas organizações. Apesar do estudo citar este desafio o autor não trabalhou ou os evidenciou no corpo do trabalho.

4.5.2 Quais são as oportunidades?

Ao analisar as oportunidades listadas nos estudos (Tabela 9), percebeu-se que os atributos trazidos transcendem o âmbito educacional, porém este estudo se limitou a analisar e ligar as oportunidades referentes à educação para não fugir do escopo de trabalho. Na tabela 9 é possível observar que entre as oportunidades mais citadas estão a facilidade de uso (com 16 citações) e o auxílio no aprendizado (com 13 citações). Ainda foram citadas algumas qualidades explicitamente ligadas à educação na sala de aula, como: acessibilidade, auxílio na entrega dos projetos em sala de aula, auxílio no treinamento, engajamento no desenvolvimento de software, diminuição do gap existente entre ciência da computação e educação, multidisciplinariedade, preparo para o mundo real e uso em diferentes níveis educacionais.

A facilidade de uso do Low Code é relacionada a um benefício explorado na terceira questão: o alto nível de abstração. O estudo [EP6], no qual o Mendix foi utilizado para desenvolvimento e apoio de cenários nos cursos de sistemas de informação, cita que, além da facilidade de uso, algumas outras oportunidades são relacionadas, como o engajamento no desenvolvimento, a entrega rápida, o foco em resultado, foco em requisitos, foco no entendimento do problema e a facilidade de uso. Nesse estudo, foi possível observar que muitos estudantes participaram do desenvolvimento e das atividades, constatando um maior engajamento desses alunos na sala de aula, uma vez que a competência de codificação não é tão requerida, ou seja, é fácil de usar. Dessa forma, com o software sendo construído de maneira facilitada e rápida, os alunos conseguiram focar em reuniões com stakeholders reais para entendimento do problema, visto que uma vantagem trazida nesse estudo é a simulação do ambiente de trabalho.

Tabela 9 — Oportunidades relacionadas ao uso do Low Code no âmbito educacional.

Oportunidades	Estudos
Acessibilidade	EP27
Aplicações Intuitivas e Simples	EP35
Auxílio na Entrega do Produto	EP11
Auxílio na Simulação	EP4
Auxílio no Aprendizado	EP5, EP6, EP8, EP10, EP14, EP18, EP25, EP26, EP32, EP33, EP34, EP36, EP37
Auxílio no Treinamento	EP10, EP14, EP16
Auxílio no Uso	EP27
Engajamento no Desenvolvimento	EP6, EP17
Entrega Rápida do Produto	EP6, EP21, EP32, EP37, EP38
Escalabilidade	EP22
Experiência de Usuário	EP38
Facilidade de Uso	EP6, EP8, EP16, EP18, EP21, EP22, EP24, EP25, EP30, EP32, EP33, EP35, EP36, EP37, EP38, EP39
Flexibilidade	EP22
Foco em Requisitos	EP6
Foco em Resultado	EP6
Foco no Entendimento do Problema	EP6
Independência do Desenvolvedor	EP39
Inovação em Soluções	EP8, EP18, EP36, EP39
Integração de Dados	EP21
Interface Gráfica	EP8, EP10, EP36, EP37, EP38
Lacuna Entre Ciência da Computação e Educação	EP2
Low Code e Modos de Trabalho	EP3
Multidisciplinariedade	EP18, EP21, EP33
Múltiplas ferramenta	EP21, EP22
Oportunidade de Pesquisa	EP12
Preparo Para o Mundo Real	EP17
Transformação Digital	EP19
Uso em Diferentes Áreas do Conhecimento	EP5
Uso em Diferentes Níveis Educacionais	EP5

Fonte: O autor (2022)

Referente ao auxílio no aprendizado, o estudo [EP36] mostrou o impacto de tecnologias emergentes como o Low Code em cursos relacionados a ciências contábeis, evidenciando esta oportunidade. Nele, o autor discute que não é o foco desenvolver técnicas em desenvolvimento de software, mas sim expor o estudante a diferentes possibilidades que poderiam vir por meio dessas tecnologias para que diferentes problemas possam ser resolvidos.

O estudo [EP37] utiliza o Low Code como instrumento para aprendizado de programação no ensino médio, principalmente por ser ter interface gráfica, ter entrega rápida e facilidade de uso, o que auxilia os alunos ao aprendizado da lógica e do processo de criação de software em si. Tudo isso aplicado na engenharia de software pode trazer benefícios também para alunos experientes, que podem se beneficiar do uso de pouco código e da facilidade de uso para focar mais na base de resolução de problemas [EP32]. Um outro exemplo interessante é o foco no aprendizado de metodologias ágeis em vez de código, que foi relatado no estudo [EP25]. Na engenharia, essa tecnologia é ótima, pois fornece meios para treinar engenheiros focando mais em processos do que código em si [EP14].

Um bom conceito trazido pelo estudo [EP17], é que com o uso de ferramentas de rápido desenvolvimento de código, é adicionado a oportunidade dos alunos em participar de todas as fases de um projeto, desde sua concepção até sua entrega, preparando-os para resolver desafios do mundo real, o que também auxilia no aprendizado de forma direta.

O estudo [EP2] introduz uma boa vantagem: a ferramenta Low Code para resolução de uma lacuna entre ciência da computação e a educação. Nele, o autor traz a problemática da falta de sincronia entre o conhecimento de banco de dados, lógica de programação, experiência de usuário, segurança e integrações. Geralmente esses conceitos são construídos em diferentes períodos do curso, dificultando a integração dos mesmos, visto que na rotina de um profissional dessa área é exigido que todos os conceitos sejam integrados o tempo todo. Então, dessa forma, o autor propõe o uso da ferramenta Mendix, pois essa ferramenta pode introduzir esses conceitos desde o início do curso de forma aplicada, permitindo que os alunos se beneficiem dessas funcionalidades antes de serem apresentados à teoria propriamente dita.

O estudo [EP39], que traz uma discussão sobre a inovação para o ambiente organizacional por meio do Low Code, defende que este tipo de ferramenta está totalmente ligado com a oportunidade de inovação, uma vez que seu uso é facilitado, trazendo mais independência para o desenvolvedor. Isso deve-se ao fato de que, em algumas ferramentas, o desenvolvedor é independente de outros profissionais para criar aplicações modelos.

5 CONCLUSÃO

Com base na responsividade das perguntas propostas e nas discussões trazidas por esta pesquisa, é possível dizer que o presente estudo obteve sucesso no mapeamento do uso do Low Code no âmbito educacional, trazendo um apurado de informações para o entendimento de uma ferramenta que tem potencial e que está ganhando o mercado de trabalho com uma grande velocidade. Portanto, se faz necessário que esta seja cada vez mais estudada e utilizada para melhorar a competitividade e as oportunidades de empregabilidade dos futuros profissionais da área de TIC.

5.1 RETOMANDO AS QUESTÕES DE PESQUISA

No geral, pode-se perceber que, o Low Code pode ser inserido em diferentes contextos, níveis educacionais e disciplinas. Que seu uso, apesar de ser maior no ensino superior e nas disciplinas relacionadas a TIC, pode variar de ensino médio até educação em diferentes contextos, como pesquisa, treinamento de pessoas com ou sem conhecimento em programação, como no caso do treinamento para pessoas desempregadas da área de STEAM, agricultores e pessoas da terceira idade.

Em relação ao seu uso, foi possível visualizar o Low Code sendo coerente com sua principal finalidade, que é o uso da ferramenta para o desenvolvimento de aplicações com auxílio dos métodos ágeis. Assim como foi identificado que esta tecnologia está adequada ao treinamento de pessoas desempregadas e para estimular o desenvolvimento de soluções inovadoras através do conceito de cidadão desenvolvedor. O desenvolvimento de algoritmos de IA também foi apontado como benefício do Low Code, possibilitado pelas bibliotecas e ferramentas com os mesmos princípios, mostrando uma versatilidade muito grande de seu uso e apontando uma miríade de novos instrumentos e usos em diversos aspectos da educação.

Para a formação profissional, o Low Code também demonstrou ser promissor com os benefícios e vantagens identificadas. Foi possível visualizar que as vantagens esperadas pela ferramenta em si foram confirmadas, mas que também foi possível identificar vantagens específicas para o apoio educacional, como o uso da ferramenta Low code para o estudo de metodologias ágeis e métodos como BDD e TDD. A facilidade do Low Code foi refletida no alto índice da resposta para a produtividade no desenvolvimento de software, o que demonstrou ser bastante presente nos estudos. Em relação a métodos ativos de aprendizado, o

Low Code também possibilita uma simulação do ambiente de trabalho, que junto com o aprendizado do Low Code, pode auxiliar na empregabilidade. Nesse aspecto, o Low Code se mostrou adaptado ao dia a dia de um desenvolvedor, com menos foco no código, possibilitando que o estudante possa participar de reuniões de resolução de problemas em todas as fases do processo.

Sobre as oportunidades e desafios, o Low Code se mostrou uma promissora ferramenta para ser utilizada na educação, trazendo oportunidades relativas ao uso de uma ferramenta fácil, acessível, flexível e multidisciplinar. Como por exemplo, a exploração de conceitos do mundo real no desenvolvimento de software, o uso da ferramenta na pesquisa, na aprendizagem da lógica de programação no ensino médio e a quebra entre as lacunas da ciência da computação com a educação. Assim, beneficiando alunos com padrões de código facilitado, possibilitando mais interação entre stakeholders de projetos que simulam um projeto do mundo real. Além disso, o Low Code mostrou contribuições em diversos panoramas do desenvolvimento de software relacionados ao conceito de Cidadão desenvolvedor, facilitado a inovação por parte de diversas categorias da sociedade.

Apesar de todos os benefícios citados ainda existem desafios, limitações e desvantagens que precisam ser enfrentados para que o Low Code se encaixe bem em no uso proposto pelo professor ou orientador no âmbito educacional. Vale destacar que esta ferramenta pode comprometer o backlog do projeto, caso o time não esteja engajado, o que em um ambiente de sala de aula pode ser desafiador, tanto para os alunos, como para o professor, que pode se tornar *product owner* do projeto, como aconteceu em alguns estudos desse mapeamento. Outros desafios, limitações e desvantagens identificadas foram: dificuldade de domínio e aprendizado por parte de alguns alunos; grande volume de dados, que pode comprometer o uso das aplicações através de conexões lentas ou desenvolvimento off-line; os modelos de precificação das licenças não são uniformes, podendo gerar dificuldades em orçamentos por parte das instituições; limitações em relação as soluções que não são alcançáveis apenas com Low Code, sendo necessário a utilização de métodos tradicionais do desenvolvimento de software; pouca qualidade de inovação nos ambientes; poucas funcionalidades pré-construídas, dificultando o aprendizado dos alunos com pouca expertise; e a portabilidade de código inexistente, fazendo com que uma aplicação tenha que ser construída do zero entre as diferentes ferramentas.

5.2 LIMITAÇÕES E AMEAÇAS DA PESQUISA

A principal limitação do estudo é relacionada ao tema, que ao ser buscado nas bases de dados, apresentou pequenas quantidades de estudos. Apenas o Google Acadêmico apresentou uma maior quantidade de estudos, porém, nem sempre estudos de qualidade. Esse aspecto é destacado como uma ameaça, visto que, por causa disso, há possibilidade de estudos relevantes que não tenham sido selecionados durante a coleta de estudos.

5.3 TRABALHOS FUTUROS

Por meio desse mapeamento, é esperado que novos trabalhos na área de Low Code e educação sejam explorados, principalmente do seu impacto para auxiliar no aprendizado, na resolução de problemas e na criação de inovação. Como também a exploração dessa ferramenta em diferentes cursos e níveis educacionais, que devem passar pela transformação digital, como por exemplo os cursos da área de STEAM, cursos de treinamento e cursos no ambiente profissional.

REFERÊNCIAS

- 14th Annual State Of Agile Report. **Digital.ai**, 28, maio 2020. Disponível em <<https://digital.ai/catalyst-blog/the-14th-annual-state-of-agile-report>> Acesso em: 19, abril 2022.
- ALFONSO, Maria Isabel; BOTIA, Antonio. An iterative and agile process model for teaching software engineering. In: **18th Conference on Software Engineering Education & Training (CSEET'05)**. IEEE, 2005. p. 9-16.
- ATKINS, Betsy. The Most Disruptive Trend Of 2021: No Code / Low Code. **Forbes**, 24, novembro 2020. Disponível em <<https://www.forbes.com/sites/betsyatkins/2020/11/24/the-most-disruptive-trend-of-2021-no-code--low-code/?sh=166581e46570>> Acesso em: 11, maio 2022.
- BARROWS, Howard S. Problem-based learning in medicine and beyond: A brief overview. **New directions for teaching and learning**, v. 1996, n. 68, p. 3-12, 1996.
- BAUMGARTEN, Christoph; SIMEON, Alex; WILHELM, Michael C. Citizen Developers Driving the Digital Campus. **European Journal of Higher Education IT**, v. 1, 2020.
- BECK, Kent *et al.* Manifesto For Agile Software Development. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 19, abril 2022.
- BLUMENFELD, Phyllis et al. Creating usable innovations in systemic reform: Scaling up technology-embedded project-based science in urban schools. **Educational psychologist**, v. 35, n. 3, p. 149-164, 2000.
- BROME, Cynthia. Active learning. **Vanderbilt University Center for Teaching**, 2016. Disponível em: < <https://cft.vanderbilt.edu/guides-sub-pages/active-learning/>>. Acesso em: 19, abril 2022.
- BRASIL. [Constituição (1988)]. **Constituição da República Federativa do Brasil**: promulgada em 5 de outubro de 1988. 4. ed. São Paulo: Saraiva, 1990.
- CARVALHO, Dárlinton Barbosa Feres et al. Promoting Digital Entrepreneurship to mitigate the impacts caused by the COVID-19 pandemic. In: **XVII Brazilian Symposium on Information Systems**. 2021. p. 1-8.
- COCCO, Sandra. Student leadership development: The contribution of project-based learning. **Unpublished Master's thesis. Royal Roads University, Victoria, BC**, 2006.
- COCKBURN, Alistair; HIGHSMITH, Jim. Agile software development, the people factor. **Computer**, v. 34, n. 11, p. 131-133, 2001.

DEWI, Deshinta Arrova; MUNIANDY, Mohana. The agility of agile methodology for teaching and learning activities. In: **2014 8th. Malaysian Software Engineering Conference (MySEC)**. IEEE, 2014. p. 255-259.

DIESEL, Aline; BALDEZ, Alda Leila Santos; MARTINS, Silvana Neumann. Os princípios das metodologias ativas de ensino: uma abordagem teórica. **Revista Thema**, v. 14, n. 1, p. 268-288, 2017.

DOS SANTOS, Simone C. et al. Two decades of PBL in teaching computing: a systematic mapping study. **IEEE Transactions on Education**, v. 64, n. 3, p. 233-244, 2020.

DRAFT, Strawman. Computer Science Curricula 2013. **ACM and IEEE Computer Society, Incorporated: New York, NY, USA**, 2013.

FERNANDES, João Paulo; ARAÚJO, Ricardo; ZENHA-RELA, Mário. Achieving Scalability in Project Based Learning through a Low-Code platform. In: **Proceedings of the 34th Brazilian Symposium on Software Engineering**. 2020. p. 710-719.

Gartner Glossary: Citizen Developer. **Gartner**. Disponível em: <<https://www.gartner.com/en/information-technology/glossary/citizen-developer>>. Acesso em: 19, abril 2022.

Global Low-Code Development Platform Market By Component, By Application, By Deployment Type, By End User, By Region, Industry Analysis and Forecast, 2020 – 2026. **Marqual IT Solutions Pvt. Ltd (KBV Research)**, maio 2020. Disponível em <<https://www.researchandmarkets.com/reports/5094295/global-low-code-development-platform-market-by>>. Acesso em: 19, abril 2022.

KITCHENHAM, Barbara; CHARTERS, Stuart. **Guidelines for performing systematic literature reviews in software engineering**. 2007.

KOKOTSAKI, Dimitra; MENZIES, Victoria; WIGGINS, Andy. Project-based learning: A review of the literature. **Improving schools**, v. 19, n. 3, p. 267-277, 2016.

LASAUSKIENE, Jolanta; RAUDUVAITE, Asta. Project-based learning at university: Teaching experiences of lecturers. **Procedia-Social and Behavioral Sciences**, v. 197, p. 788-792, 2015.

LITMAN, Michael; FIELD, Daniela. Mendix as a solution for present gaps in Computer Programming in Higher Education. 2018.

MEIJERS, Jesse. The difference between low-code and no-code platforms. **Triggre**. Disponível em <<https://www.triggre.com/blog/the-difference-between-low-code-and-no-code->

[platforms#:~:text=Low%2Dcode%20platforms%20are%20aimed,other%20hand%2C%20target%20business%20users.>](#) Acesso em: 11, maio 2022.

METRÔLHO, José Carlos; RIBEIRO, Fernando Reinaldo; PASSÃO, Pedro. Teaching Agile Software Engineering Practices Using Scrum and a Low-Code Development Platform—A Case Study. **ICSEA 2020**, p. 170, 2020.

MEW, Lionel; FIELD, Daniela. The Course Registration App: A Low Code Development Scenario to Support Core IS Courses. **Proceedings of the EDSIG Conference**, v. 4, 2018.

MIODUSER, David; BETZER, Nadav. The contribution of Project-based-learning to high-achievers' acquisition of technological knowledge and skills. **International Journal of technology and design education**, v. 18, n. 1, p. 59-77, 2008.

OLIVEIRA, Armanda Maria C. Amorim; DOS SANTOS, Simone C.; GARCIA, Vinicius Cardoso. PBL in teaching computing: An overview of the last 15 years. In: **2013 IEEE Frontiers in Education Conference (FIE)**. IEEE, 2013. p. 267-272.

OUTSYSTEMS. Extend Logic with Your Own Code. **Outsystems**, 6, maio 2022.

Disponível em: <

https://success.outsystems.com/Documentation/11/Extensibility_and_Integration/Extend_Logic_with_Your_Own_Code>. Acesso em: 16, maio 2022.

RICHARDSON, Clay *et al.* New development platforms emerge for customer-facing applications. Forrester: **Cambridge**, MA, USA, v. 15, 2014.

RIMOL, Meghan. Gartner Says the Majority of Technology Products and Services Will Be Built by Professionals Outside of IT by 2024 **Gartner**, Stamford, 14, junho 2021.

Disponível em: <<https://www.gartner.com/en/newsroom/press-releases/2021-06-10-gartner-says-the-majority-of-technology-products-and-services-will-be-built-by-professionals-outside-of-it-by-2024>>. Acesso em: 19, abril 2022.

SALZA, Pasquale; MUSMARRA, Paolo; FERRUCCI, Filomena. Agile methodologies in education: A review. **Agile and lean concepts for teaching and learning**, p. 25-45, 2019.

SAVERY, John R. Overview of problem-based learning: Definitions and distinctions. **Essential readings in problem-based learning: Exploring and extending the legacy of Howard S. Barrows**, v. 9, n. 2, p. 5-15, 2015.

THACKER, Donald *et al.* Business Students as Citizen Developers: Assessing Technological Self-Conception and Readiness. **INFORMATION SYSTEMS EDUCATION JOURNAL**, v. 19, n. 5, p. 5, 2021.

The State of Application Development: Is IT Ready for Disruption? **OutSystems**, 2019. Disponível em <<https://www.outsystems.com/1/state-app-development-insurance/>>. Acesso em: 19, abril 2022.

TUCKER, Allen B. Strategic directions in computer science education. **ACM Computing Surveys (CSUR)**, v. 28, n. 4, p. 836-845, 1996.

VINCENT, Paul *et al.* Industry Report: Magic Quadrant for Enterprise High Productivity Application Platform as a Service. **Gartner report**, 2018.

VINCENT, Paul *et al.* Magic Quadrant for Enterprise Low-Code Application Platforms. **Gartner report**, 2020.

VINCENT, Paul *et al.* Magic Quadrant for Enterprise Low-Code Application Platforms. **Gartner report**, 2021.

WOHLIN, Claes. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: **Proceedings of the 18th international conference on evaluation and assessment in software engineering**. 2014. p. 1-10.

WOODS Donald. Problem based learning: How to get the most from PBL. **McMaster University**, 1994.

YAMPOLSKY, Mark; SCACCHI, Walt. Learning game design and software engineering through a game prototyping experience: A case study. In: **2016 IEEE/ACM 5th International Workshop on Games and Software Engineering (GAS)**. IEEE, 2016. p. 15-21.

APÊNDICE A — Estudos primários identificados nesse mapeamento.

[EP1] ISHIKAWA, Edison; RALHA, Célia Ghedini. Uma Abordagem Prática com Foco em Gestão de TIC para Ensino de SI para Alunos de Graduação em Computação. **iSys-Brazilian Journal of Information Systems**, v. 9, n. 2, 2016.

[EP2] LITMAN, Michael; FIELD, Daniela. Mendix as a solution for present gaps in Computer Programming in Higher Education. **Twenty-fourth Americas Conference on Information Systems**, 2018.

[EP3] MEW, Lionel; FIELD, Daniela. A case study on using the mendix low code platform to support a project management course. In: **Proceedings of the EDSIG Conference ISSN**. 2018. p. 3857.

[EP4] MCHANEY, Roger. Simulation education in non-simulation courses. In: **2018 Winter Simulation Conference (WSC)**. IEEE, 2018. p. 4038-4045.

[EP5] BAGGIA, ALENKA et al. Oracle APEX in Higher Education. **37th International Conference on Organizational Science Development**, 2018.

[EP6] MEW, Lionel; FIELD, Daniela. The Course Registration App: A Low Code Development Scenario to Support Core IS Courses. **Proceedings of the EDSIG Conference**, v. 4, 2018.

[EP7] METRÔLHO, J. C. M. M.; RIBEIRO, Fernando Reinaldo. Software Engineering Education: Sharing an approach, experiences, survey and lessons learned. **ICSEA 2018**, p. 89, 2018.

[EP8] HERRIMAN, Seamus H. **Development and Testing of a Curriculum for an Introductory Course for Makerspace in High School**. 2018. Tese de Doutorado. College of William and Mary

[EP9] BAGGIA, Alenka; LESKOVAR, Robert; RODIČ, Blaž. Low code programming with oracle APEX offers new opportunities in higher education. In: **third International Scientific Conference ITEMA Recent Advances in Information Technology, Tourism, Economics, Management and Agriculture**. 2019.

[EP10] METRÔLHO, J. C. M. M. et al. An approach using a low-code platform for retraining professionals to ICT. In: **11th International Conference on Education and New Learning Technologies**. IATED, 2019. p. 7200-7207.

- [EP11] POE, Laura F.; MEW, Lionel. Implementing Agile as an Instructional Methodology for Low-Code Software Development Courses. In: **Proceedings of the EDSIG Conference ISSN**. 2019. p. 4901.
- [EP12] FERNANDES, João Paulo; ARAÚJO, Ricardo; ZENHA-RELA, Mário. Achieving Scalability in Project Based Learning through a Low-Code platform. In: **Proceedings of the 34th Brazilian Symposium on Software Engineering**. 2020. p. 710-719.
- [EP13] ROSA, Valéria Argôlo; DE SOUZA MATOS, Ecivaldo. Strategies to improve engagement of long-lived elderly women (80+) in the interaction co-design process: challenges and lessons learned. In: **Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems**. 2020. p. 1-10.
- [EP14] ADRIAN, Benjamin; HINRICHSEN, Sven; NIKOLENKO, Alexander. App development via Low-Code programming as part of modern industrial engineering education. In: **International Conference on Applied Human Factors and Ergonomics**. Springer, Cham, 2020. p. 45-51.
- [EP15] DANIEL, Gwendal et al. Xatkit: a multimodal low-code chatbot development framework. **IEEE Access**, v. 8, p. 15332-15346, 2020.
- [EP16] METRÔLHO, J. C. M. M.; RIBEIRO, F.; ARAÚJO, Ricardo. A strategy for facing new employability trends using a low-code development platform. In: **14th International Technology, Education and Development Conference**. 2020. p. 8601-8606.
- [EP17] CRUMBLY, Jack; FIELD, Daniela. Implementing No/Low Code Development in a Systems Analysis and Design Course. **Journal of Strategic Innovation and Sustainability**, v. 15, n. 5, p. 116-130, 2020.
- [EP18] THACKER, Donald et al. Assessing Technological Self-Conception: Are Business Students Ready to be Citizen Developers?. In: **Proceedings of the EDSIG Conference ISSN**. 2020. p. 4901.
- [EP19] BAUMGARTEN, Christoph; SIMEON, Alex; WILHELM, Michael C. Citizen Developers Driving the Digital Campus. **European Journal of Higher Education IT**, v. 1, 2020.
- [EP20] YATES, David J.; WAGUESPACK, Leslie J.; BABB, Jeffrey S. Reflections from Recent Graduates on a Computer Information Systems Culminating Experience. In: **Proceedings of the EDSIG Conference ISSN**. 2020. p. 4901.

- [EP21] METRÔLHO, José Carlos; RIBEIRO, Fernando Reinaldo; PASSÃO, Pedro. Teaching Agile Software Engineering Practices Using Scrum and a Low-Code Development Platform—A Case Study. **ICSEA 2020**, p. 170, 2020.
- [EP22] KARIĆ, Katarina; RADOVIĆ, Maja; MITROVIĆ, Katarina. Development and Evaluation of CATAPEX Tutorial for the Database Practice Course. **8th International Scientific Conference Technics and Informatics in Education**, 2020
- [EP23] CARVALHO, Dárlinton Barbosa Feres et al. Promoting Digital Entrepreneurship to mitigate the impacts caused by the COVID-19 pandemic. In: **XVII Brazilian Symposium on Information Systems**. 2021. p. 1-8.
- [EP24] OTEYO, Isaac Nyabisa et al. Building Smart Agriculture Applications Using Low-Code Tools: The Case for DisCoPar. In: **2021 IEEE AFRICON**. IEEE, 2021. p. 1-6.
- [EP25] LEBENS, Mary; FINNEGAN, Roger. Using a Low Code Development Environment to Teach the Agile Methodology. In: **International Conference on Agile Software Development**. Springer, Cham, 2021. p. 191-199.
- [EP26] POE, Laura; MEW, Lionel. The effects of using the agile methodology as an instructional format for software development courses. **Industry and Higher Education**, p. 09504222211058658, 2021.
- [EP27] VILLEGAS-CH, William; GARCÍA-ORTIZ, Joselin; SÁNCHEZ-VITERI, Santiago. Identification of the Factors That Influence University Learning with Low-Code/No-Code Artificial Intelligence Techniques. **Electronics**, v. 10, n. 10, p. 1192, 2021.
- [EP28] KABATHOVA, Janka; DRLIK, Martin. Towards predicting student's dropout in university courses using different machine learning techniques. **Applied Sciences**, v. 11, n. 7, p. 3130, 2021.
- [EP29] HOW, Meng-Leong et al. Artificial Intelligence for Social Good in Responsible Global Citizenship Education: An Inclusive Democratized Low-Code Approach. In: **Proc. of the 3rd World Conference on Teaching and Education**. 2021. p. 81-89.
- [EP30] ABIDIN, Azwan; SENIN, Noorhaslina; MANAF, Ahmad Azaini Abdul. A PRELIMINARY STUDY OF LOW-CODE/NO-CODE ECOSYSTEM PRACTICES: TRANSLATING DESIGN STUDENT VIEWS ON CRAFTING INTERACTIVE DESIGN. **NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal| NVEO**, p. 10244-10258, 2021.

- [EP31] DIERL, Simon et al. Do Away with the Frankensteinian Programs! A Proposal for a Genuine SE Education. In: **2021 Third International Workshop on Software Engineering Education for the Next Generation (SEENG)**. IEEE, 2021. p. 26-30.
- [EP32] ELLIS, Geraghty Anne; MULKA, Nicholas; JARIWALA, Amit Shashikant. Rapid Development of Software Solutions to Enhance Course Infrastructure and the Educational Experience of Student Developers. In: **2021 ASEE Virtual Annual Conference Content Access**. 2021.
- [EP33] MATOOK, Sabine et al. Experiential learning in Work-Integrated Learning (WIL) projects for metacognition: integrating theory with practice. 2021. **Australasian Conference on Information Systems**, 2021
- [EP34] ARNOLD, Patricia et al. Scaffolding International Interdisciplinary Innovation Challenges in Higher Education. **International Conference The Future of Education**. 2021.
- [EP35] KERPEN, Daniel et al. COMBINING COLLABORATIVE USER EXPERIENCE DESIGN WITH CROWD ENGINEERING: A PROBLEM-BASED LAB COURSE FOR (UNDER-) GRADUATE STUDENTS. In: **DS 110: Proceedings of the 23rd International Conference on Engineering and Product Design Education (E&PDE 2021), VIA Design, VIA University in Herning, Denmark. 9th-10th September 2021**. 2021.
- [EP36] WANG, Tawei. The impact of emerging technologies on accounting curriculum and the accounting profession. **Pacific Accounting Review**, 2021.
- [EP37] FINKLER, Gustavo Hanke et al. ESTUDO DE SISTEMAS DE DESENVOLVIMENTO DE MOBILE APPS PARA ENSINO DE PROGRAMAÇÃO NA REDE BÁSICA DE ENSINO. **Salão do Conhecimento**, v. 7, n. 7, 2021.
- [EP38] DA SILVA JUNIOR, Braz Araujo; DA COSTA CAVALHEIRO, Simone André; FOSS, Luciana. GameStation: Specifying Games with Graphs. In: **Anais do XXXII Simpósio Brasileiro de Informática na Educação**. SBC, 2021. p. 499-511.
- [EP39] KREJCI, Désirée; IHO, Satu; MISSIONIER, Stéphanie. Innovating with employees: an exploratory study of idea development on low-code development platforms. 2021. **Association for Information Systems**. 2021

APÊNDICE B — Índice de responsividade dos estudos para as questões de pesquisa

ID	Q1	Q2	Q3	Q4	Total
EP1	1,0	1,0	1,0	0,0	3,0
EP2	1,0	1,0	1,0	0,5	3,5
EP3	1,0	1,0	1,0	1,0	4,0
EP4	1,0	0,5	1,0	0,5	3,0
EP5	1,0	1,0	1,0	1,0	4,0
EP6	1,0	1,0	1,0	1,0	4,0
EP7	1,0	1,0	1,0	0,0	3,0
EP8	1,0	1,0	1,0	1,0	4,0
EP9	1,0	1,0	1,0	0,0	3,0
EP10	1,0	1,0	1,0	1,0	4,0
EP11	1,0	1,0	1,0	1,0	4,0
EP12	1,0	1,0	1,0	1,0	4,0
EP13	1,0	0,5	0,0	0,0	1,5
EP14	1,0	0,5	1,0	0,5	3,0
EP15	1,0	1,0	1,0	0,0	3,0
EP16	1,0	1,0	1,0	1,0	4,0
EP17	1,0	1,0	1,0	0,5	3,5
EP18	1,0	0,5	1,0	1,0	3,5
EP19	1,0	1,0	1,0	1,0	4,0
EP20	1,0	1,0	0,0	0,5	2,5
EP21	1,0	1,0	1,0	1,0	4,0
EP22	1,0	1,0	0,0	1,0	3,0
EP23	0,5	0,5	0,5	0,0	1,5
EP24	0,5	1,0	1,0	1,0	3,5
EP25	1,0	1,0	1,0	0,5	3,5
EP26	1,0	1,0	1,0	0,5	3,5
EP27	1,0	1,0	0,0	0,5	2,5
EP28	1,0	0,5	0,0	0,0	1,5
EP29	0,5	0,5	1,0	0,0	2,0
EP30	1,0	0,5	1,0	1,0	3,5
EP31	1,0	0,5	0,5	0,0	2,0
EP32	1,0	0,5	1,0	1,0	3,5
EP33	1,0	1,0	1,0	1,0	4,0

EP34	1,0	1,0	1,0	0,5	3,5
EP35	1,0	1,0	0,0	0,5	2,5
EP36	1,0	0,0	1,0	1,0	3,0
EP37	1,0	0,5	1,0	1,0	3,5
EP38	1,0	0,5	1,0	1,0	3,5
EP39	1,0	0,5	1,0	1,0	3,5
Total	37,5/39	31,5/39	32,0/39	25,0/39	Média: 3,5
Responsividade (%)	96,15	80,77	82,05	64,10	81,41
