



UEPB

**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I – CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM COMPUTAÇÃO**

BRUNO SILVA DE ARAÚJO

**LOW-CODE: UMA PESQUISA SOBRE UMA DAS ABORDAGENS DE
DESENVOLVIMENTO DE SOFTWARE**

**CAMPINA GRANDE
2023**

BRUNO SILVA DE ARAÚJO

**LOW-CODE: UMA PESQUISA SOBRE UMA DAS ABORDAGENS DE
DESENVOLVIMENTO DE SOFTWARE**

Trabalho de Conclusão de Curso (Artigo) apresentado ao Departamento do Curso de Computação da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Computação.

Orientador: Prof. Me. Edson Holanda Cavalcante Júnior.

**CAMPINA GRANDE
2023**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

A663I Araújo, Bruno Silva de.
Low-code [manuscrito] : uma pesquisa sobre uma das abordagens de desenvolvimento de software / Bruno Silva de Araujo. - 2023.
24 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2023.

"Orientação : Prof. Me. Edson Holanda Cavalcante Júnior, Coordenação do Curso de Computação - CCT. "

1. Low-Code. 2. Desenvolvimento de software. 3. Programação computacional. I. Título

21. ed. CDD 005.1

BRUNO SILVA DE ARAÚJO

LOW-CODE: UMA PESQUISA SOBRE UMA DAS ABORDAGENS DE
DESENVOLVIMENTO DE SOFTWARE

Trabalho de Conclusão de Curso (Artigo)
apresentado ao Departamento do Curso
de Computação da Universidade Estadual
da Paraíba, como requisito parcial à
obtenção do título de bacharel em
Computação.

Aprovada em: 03/03/2023.

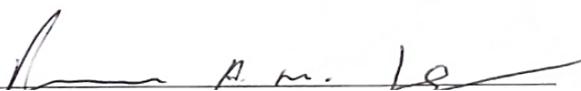
BANCA EXAMINADORA



Prof. Me. Edson Holanda Cavalcante Júnior (Orientador)
Universidade Estadual da Paraíba (UEPB)



Prof. Dr. Wellington Candeia de Araújo
Universidade Estadual da Paraíba (UEPB)



Prof. Me. Francisco Anderson Mariano da Silva
Universidade Estadual da Paraíba (UEPB)

A minha família, pelo apoio,
companheirismo e amor, DEDICO.

“Por mais brilhante que a estratégia seja,
você deve sempre olhar para os
resultados.”

Winston Churchill

LISTA DE ILUSTRAÇÕES

Figura 1 –	Motivos para usar as plataformas <i>Low-Code</i>	10
Figura 2 –	Projetos que estão usando <i>low-code</i>	10
Figura 3 –	Tela de criação do aplicativo Lightning.....	13
Figura 4 –	Opções do aplicativo.....	14
Figura 5 –	Utilitários.....	14
Figura 6 –	Itens de navegação.....	15
Figura 7 –	Perfis de usuário.....	15
Figura 8 –	Tela de início.....	16
Figura 9 –	Leads.....	16
Figura 10 –	Contas.....	16
Figura 11 –	Contatos.....	17
Figura 12 –	Oportunidades.....	17
Figura 13 –	Chatter.....	17

SUMÁRIO

1	INTRODUÇÃO	8
2	REFERENCIAL TEÓRICO	9
2.1	Low-Code	9
2.1.1	<i>Funcionamento da abordagem low-code</i>	11
2.1.2	<i>Low-code x programação tradicional</i>	11
2.2	Plataformas de desenvolvimento low-code	11
2.2.1	<i>Microsoft PowerApps</i>	12
2.2.2	<i>OutSystems</i>	12
2.2.3	<i>Mendix</i>	12
2.2.4	<i>Salesforce Lightning AppBuilder</i>	12
2.2.5	<i>Google AppSheet</i>	12
3	METODOLOGIA	13
4	RESULTADOS E DISCUSSÕES	13
4.1	Criação da aplicação usando plataforma low-code	13
4.2	Relação entre programação prática e a literatura analisada	18
4.2.1	<i>Vantagens do uso do low-code</i>	18
4.2.2	<i>Desvantagens do uso do low-code</i>	19
5	CONSIDERAÇÕES FINAIS	20
	REFERÊNCIAS	21

LOW-CODE: UMA PESQUISA SOBRE UMA DAS ABORDAGENS DE DESENVOLVIMENTO DE SOFTWARE

LOW-CODE: A SURVEY OF ONE DEVELOPMENT APPROACH

ARAÚJO, Bruno Silva de*

RESUMO

Estamos imersos na era digital e, muitas vezes, somos confrontados com a necessidade de desenvolver habilidades que se tornam cada vez mais necessárias, como a programação. Tradicionalmente, essa área exigia um conhecimento técnico restrito apenas para profissionais da área de tecnologia. No entanto, ao longo dos anos, foram desenvolvidas abordagens que facilitam o acesso na criação de aplicações. Comumente associamos a programação com várias linhas de código, múltiplas funções, classes e um trabalho tedioso, que soa como algo que dificulta o entendimento das pessoas com pouco conhecimento técnico da área de programação. Devido à alta demanda de soluções rápidas, uma abordagem vem ganhando cada vez mais destaque na indústria: o desenvolvimento *low-code*, que pode se tornar uma alternativa para suprir a demanda de profissionais capacitados no mercado, permitindo a elaboração de aplicações através de programação visual (clicar e arrastar) e pela linguagem oral. Desta forma, será possível a integração de especialistas de domínio sem que tenham necessariamente o conhecimento técnico em programação para o desenvolvimento de aplicações. Para identificar as vantagens da abordagem *low-code* e refletir sobre o futuro do desenvolvimento, realizamos uma revisão bibliográfica descritiva, juntamente com uma demonstração prática. De modo geral, as discussões revelam um cenário de pouco uso prático da abordagem no Brasil, mas apresenta um amplo campo internacional, onde seus benefícios e limitações devem ser conhecidos para que o planejamento ocorra conforme o esperado.

Palavras-chave: low-code; desenvolvimento de software; programação computacional.

ABSTRACT

We are immersed in the digital age and are often faced with the need to develop skills that are becoming increasingly necessary, such as programming. Traditionally, this area required technical knowledge that was restricted only to technology professionals. However, over the years, approaches have been developed that make it easier to create applications. We commonly associate programming with several lines of code, multiple functions, classes, and tedious work, which sounds like something that makes it difficult for people with little technical knowledge in the programming area to understand. Due to the high demand for fast solutions, an approach is gaining more and more prominence in the industry: the low-code development, which can become an alternative to supply the demand for skilled professionals in the market, allowing the development of applications through visual programming (click and drag) and by

* Graduando bacharelado em Ciências da Computação pela Universidade Estadual da Paraíba – UEPB, e-mail: bruno03araujo@gmail.com.

oral language. In this way, it will be possible to integrate domain experts without necessarily having the technical knowledge in programming for application development. To identify the advantages of the low-code approach and reflect on the future of development, we conducted a descriptive literature review along with a practical demonstration. In general, the discussions reveal a scenario of little practical use of the approach in Brazil, but it presents a broad international field, where its benefits and limitations must be known in order for planning to occur as expected.

Keywords: low-code; software development; computer programming.

1 INTRODUÇÃO

O desenvolvimento de *softwares* tradicional baseado em *mainframe* por exemplo, estão perdendo espaço devido ao elevado custo para realizar alterações e correções, sendo necessário um planejamento e documentação imutável antes de iniciar sua criação, desenvolvimento e implementação. A metodologia cascata foi criada para minimizar os problemas de erros, *debugs* e falhas no projeto, a metodologia de prototipação possibilitou a criação de um protótipo capaz de assumir até três formas distintas, sendo uma metodologia mais eficiente do que a anteriormente mencionada. Posteriormente a essas metodologias, outras foram desenvolvidas, tendo abordagens para atender às diversas demandas.

Nos encontramos na era da computação, pervasiva e com isso algumas habilidades são cada vez mais necessárias, dentre elas podemos destacar a programação, que por vezes exigia mais conhecimento técnico, o que dificultava o entendimento por parte de pessoas de outras áreas, contudo, com novas abordagens de desenvolvimento que trazem facilidade e velocidade na criação de aplicações, novas oportunidades surgem até mesmo para pessoas com pouco conhecimento técnico. A abordagem de desenvolvimento *low-code* é uma forma de desenvolver aplicações, que permite o desenvolvimento de aplicações sem a necessidade de conhecer uma linguagem de programação específica, um simples clicar e arrastar, tornando simples e rápido o desenvolvimento do software.

Atualmente, o mercado de Tecnologia da Informação (TI) enfrenta dificuldades com relação à complexidade das operações e a necessidade de prover soluções rápidas que atendam aos requisitos estabelecidos pelos clientes. Segundo Hirzel (2022), a programação *low-code* usa de técnicas alternativas mais próximas de como os usuários pensam naturalmente sobre sua tarefa. Para Sanchis et al. (2020) é inquestionável que as empresas têm de dar resposta a estas diversas exigências do mercado rapidamente.

O foco das plataformas de desenvolvimento *low-code* é proporcionar o envolvimento dos usuários finais ao atendimento de suas necessidades através do desenvolvimento de softwares (SAHINSLAN, 2021). Um recente estudo conduzido pelo Gartner¹ em junho de 2021, fez uma previsão onde os profissionais de fora da tecnologia da informação serão responsáveis pela construção de cerca de 80% dos produtos e serviços de tecnologia até o ano de 2024 (RIMOL, 2021).

Partindo destas premissas mencionadas e do atual envolvimento com esta área profissional, se fez importante realizar uma pesquisa sobre a temática com o objetivo

¹ “Gartner Says the Majority of Technology Products and Services will be built by professionals outside of IT by 2024.” (RIMOL, 2021).

geral de identificar as vantagens e desvantagens tanto na teoria quanto na prática do uso dessa abordagem de desenvolvimento. Ainda será possível evidenciar o que os autores vêm discutindo sobre as diferenças entre as abordagens de desenvolvimento *low-code* e a tradicional, identificar quais os benefícios e as limitações que são apresentadas, além de apresentar uma visão geral sobre as principais plataformas de desenvolvimento *low-code* e demonstrar o passo-a-passo da criação de um aplicativo usando o criador de aplicativos *Lightning* da *Salesforce*.

2 REFERENCIAL TEÓRICO

2.1 Low-Code

O termo “*low-code*” foi criado pela empresa norte-americana de pesquisa de mercado que presta diversos serviços de assessoria sobre os impactos e potencial de tecnologia, *Forrester Research* em 2014, para designar uma plataforma de desenvolvimento de aplicações voltadas inteiramente ao cliente (SANCHIS et al., 2020).

Para desenvolver os aplicativos que estão disponíveis na *web* e os que estão disponíveis para os dispositivos é necessário tarefas tediosas, sendo estas tarefas o maior custo para a construção de um sistema inteligente bem estruturado (CHEN, 2022). Pautado em uma menção do ex-CEO do *GitHub*, Chris Wanstrath, a probabilidade para o futuro é que os designs passem a ser feitos sem “nenhum código”, logo tem-se que:

O objetivo de usar a programação de *low-code* ou *no-code* (LCNC) é automatizar a geração de código durante o desenvolvimento do aplicativo para reduzir o esforço de desenvolvimento e colocar aplicativos em produção. De acordo com a pesquisa do *Gartner*, mais de 65% das empresas adotarão tecnologia sem código para transformação digital em 2024; no mesmo tempo, a *Global Newswire* também acredita que até 2030, a taxa de crescimento composta de plataforma *no-code* atingirá 31% e a receita chegará a 180 bilhões de dólares. (CHEN, 2022, p. 5).

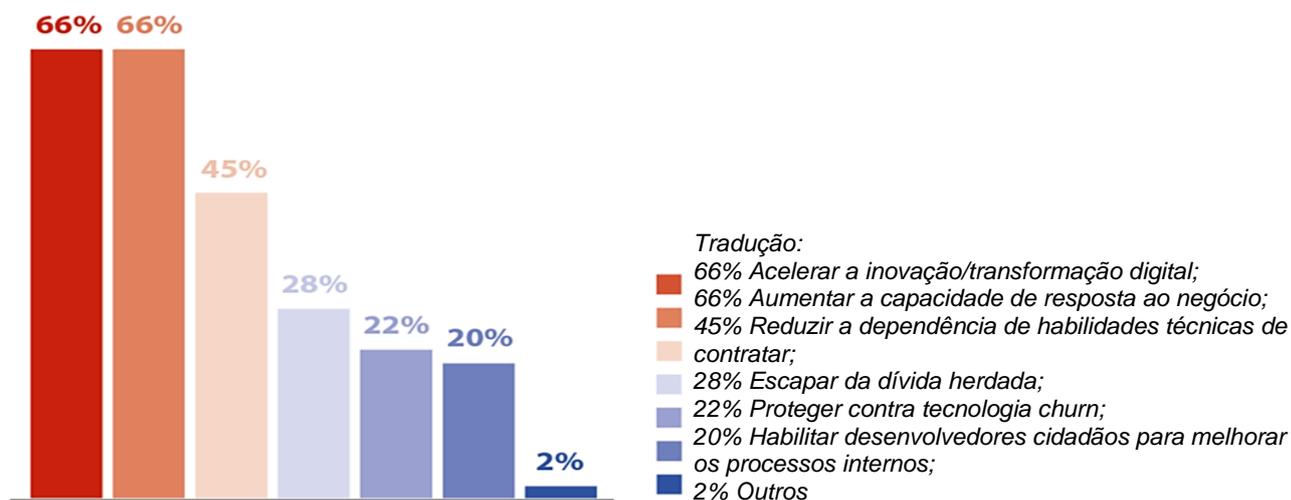
Plataformas de desenvolvimento *low-code* (*Zeev*, *Salesforce*, *Zoho*, *Mendix*, *Appian*, *Outsystems*, *Caspio*, *AppSheet*, *FileMaker*, *BuildFire*, entre outras) visam preencher a lacuna entre os desenvolvedores de software experientes e pessoas com pouca ou nenhuma experiência em Engenharia de Software (HIRWE et al., 2020). Para Waszkowski (2019), o *low-code* é um conjunto de ferramentas para programadores e também para não programadores.

Os usuários de *low-code* têm níveis que variam desde desenvolvedores profissionais até os desenvolvedores cidadãos ou voluntários, este último é um programador amador que tem pouca formação em programação e tendem a ter menos domínio sobre conhecimentos técnicos, em paralelo, as plataformas de baixo código devem elevar o nível para atender aos *pro-developers* (aqueles profissionais que tem carreira ou formação em desenvolvimento de software e trabalham com produção) (HIRZEL, 2022).

Uma pesquisa conduzida pela *OutSystems* que contou com mais de 3.300 profissionais da TI ao redor do mundo como participantes de todos os setores industriais, buscou respostas sobre os principais motivos que os levaram a usar uma plataforma *low-code*, entre as respostas as que mais ganharam destaque ou repetição, foram: acelerar a transformação digital, aumentar a capacidade de resposta

aos negócios e reduzir a dependência de habilidades técnicas difíceis de contratar. A Figura 1 representa mais dados sobre as respostas da pesquisa.

Figura 1 – Motivos para usar as plataformas *Low-Code*

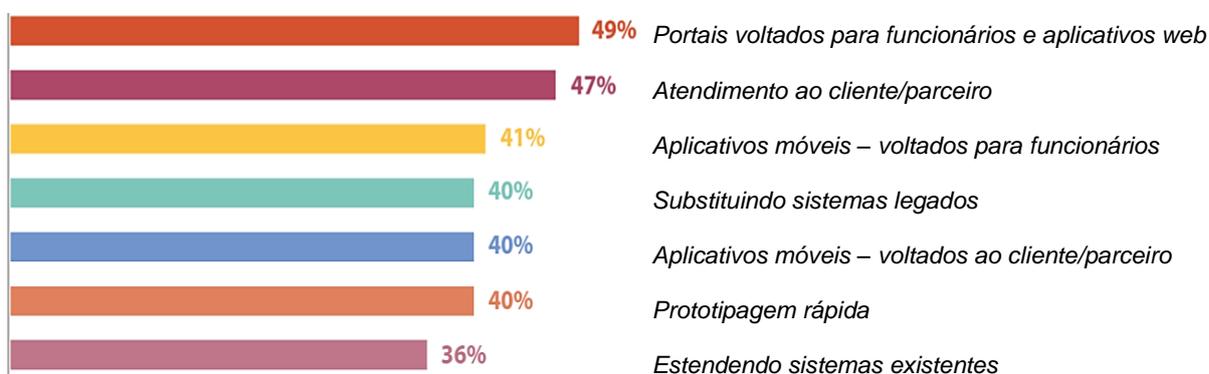


Fonte: Adaptado de *OutSystems* (2019, p. 34).

A maioria das respostas em “Outros” estão associadas às variações sobre o tema de ‘entrega mais rápida’, mesmo que alguns outros comentários mais interessantes incluíssem ‘reduzir o atraso do projeto’ ou ‘capacidade de se concentrar em requisitos de negócios importantes em vez de detalhes técnicos’, por exemplo.

Quando questionados sobre quais os tipos de projetos o *low-code* estava sendo utilizado, a maioria respondeu que estavam usando para desenvolver portais e aplicativos baseados na *web* a serem usados por funcionários e clientes ou parceiros. Outras respostas ganharam destaque como, a distribuição uniforme para aplicativos móveis (voltados para funcionários e clientes), para substituir sistemas legados, estender sistemas existentes e prototipagem rápida, conforme Figura 2.

Figura 2 – Projetos que estão usando *low-code*



Fonte: Adaptado de *OutSystems* (2019, p. 32).

2.1.1 Funcionamento da abordagem *low-code*

Low-code funciona através da utilização de uma plataforma visual que possibilita ao usuário criar suas aplicações sem escrever código manualmente. A plataforma possui ferramentas gráficas que permitem ao usuário definir a lógica e a funcionalidade da aplicação, como formulários, workflows, regras de negócio e integrações com outras fontes de dados. O *low-code* gera automaticamente o código fonte da aplicação, que pode ser personalizado por programadores quando necessário. O ponto principal do *low-code* é tornar o processo de desenvolvimento mais rápido, acessível e eficiente.

A Forrester caracteriza as plataformas *low-code* como: "produtos e/ou serviços em nuvem para desenvolvimento de aplicativos que empregam técnicas visuais e declarativas em vez de programação e estão disponíveis para clientes com baixo ou nenhum custo em dinheiro e tempo de treinamento para começar, com custos aumentando na proporção de o valor comercial das plataformas" (RICHARDSON *et al.*, 2017, p. 438 *apud* ELSHAN; DICKHAUT; EBEL, 2023).

Segundo Souza (2022), o *low-code* é um método ágil sendo o *framework* mais usado no desenvolvimento de *software*, isto pode ser identificado ao analisar o 14º Relatório Anual *State of Agile*, onde 95% dos participantes afirmam que o uso de uma metodologia ágil para desenvolvimento, culmina no aprimoramento, na melhor capacidade de gerenciamento de mudanças e de prioridades, o aumento na produtividade e uma facilidade no alinhamento dos negócios.

2.1.2 *Low-code* x programação tradicional

A diferença principal entre as duas abordagens é que na programação tradicional, o código é escrito manualmente por programadores. Na abordagem *low-code*, o código é gerado através de uma plataforma visual, que possibilita o usuário criar aplicações sem a necessidade de escrever código, isto torna o processo de desenvolvimento mais rápido e acessível a não-programadores. O desenvolvimento de *software low-code* consiste em minimizar a quantidade de codificação manual usando ferramentas gráficas (TRIGO; VARAJÃO; ALMEIDA, 2022).

Com o uso de tecnologias *low-code*, quase tudo é desenvolvido graficamente, com pouca ou nenhuma programação, permitindo que pessoas sem treinamento de programação criem aplicativos de *software*. Esses aspectos tornam o desenvolvimento de aplicativos usando *low-code* mais rápido, resultando também em uma entrega mais rápida e maior produtividade. Um fator contra essas tecnologias são os custos de licenciamento, que normalmente são mais altos em comparação com tecnologias baseadas em código (TRIGO; VARAJÃO; ALMEIDA, 2022, p. 61).

2.2 Plataformas de desenvolvimento *low-code*

Plataformas de desenvolvimento de *Low-code* são plataformas de *software* em nuvem que habilitam desenvolvedores de diferentes domínios do conhecimento e experiência técnica para desenvolver aplicativos completos pronto para produção (RICHARDSON, 2016 *apud* SAHAY *et al.*, 2020). Abaixo serão abordados alguns aspectos gerais de algumas das principais plataformas de desenvolvimento *low-code*.

2.2.1 Microsoft PowerApps

O Power Apps é um conjunto de aplicativos, serviços e conectores, bem como uma plataforma de dados que oferece um ambiente de desenvolvimento rápido de aplicativos para criação dos apps personalizados para suas necessidades de negócios. O Power Apps "democratiza" a experiência de criação de aplicativos de negócios personalizados, permitindo que os usuários criem aplicativos de negócios personalizados e repletos de recursos sem escrever código (KUMARVIVEK, 2023). Possui uma avaliação gratuita de 30 dias.

2.2.2 OutSystems

É uma plataforma de desenvolvimento *low-code* que possibilita a criação de aplicações utilizando pouco código e sempre de maneira rápida uma vez que quase toda a programação é realizada de forma visual. De forma veloz é possível criar telas e fluxos de funcionamento dos aplicativos. É possível criar aplicações *web*, *mobile* nativa além de fazer integrações com APIs, em contrapartida, a ferramenta não apresenta um ambiente que possibilita ao usuário desenvolver sem utilizar o código tradicional, isto faz com que a curva de aprendizado seja maior, outro fator limitante está em torno da licença devido ao seu alto custo, logo apenas empresas de grande porte conseguem aderir (SILVA, 2021).

A plataforma permite a conexão com bancos de dados como *Oracle*, *SqlServer*, *MySQL* e *iDB2*, isso a coloca como ferramenta compatível com o ambiente existente nas grandes empresas. Integrações com *webservice*s podem ser realizadas com poucos cliques tornando o rápido o processo de entrega que na programação convencional, com código, é bem extenso (MUNDO OUTSYSTEMS, 2019). Possui versão gratuita oferecendo menos recursos.

2.2.3 Mendix

Mendix App Services é uma coleção de alto valor de componentes de interface do usuário e blocos de construção de API que ajudam você a criar soluções sofisticadas que abordam problemas e casos de uso específicos. Esses blocos de construção expõem recursos ricos de uma forma que seus desenvolvedores (cidadãos e profissionais) podem rapidamente integrar à solução (COLLABORATIVE VISUAL DEVELOPMENT | MENDIX, 2023). Possui versão de teste gratuito.

2.2.4 Salesforce Lightning App Builder

O *Lightning App Builder* é uma ferramenta de apontar e clicar que facilita a criação de páginas personalizadas para o aplicativo móvel *Salesforce* e o *Lightning Experience*, oferecendo aos usuários o que eles precisam em um só lugar (SALESFORCE, 2023). Possui versão de teste gratuito.

2.2.5 Google AppSheet

Permite que todas as pessoas da organização criem e estendam aplicativos sem programação, combinando a Inteligência Artificial (IA) do Google e *low-code* para

criar rapidamente apps com automação, poupando tempo e talento. Seu preço é baseado no número de usuários finais, mas possibilita a criação gratuita de apps (GOOGLE CLOUD, 2023).

3 METODOLOGIA

A presente pesquisa se trata de uma revisão bibliográfica sistemática descritiva, de cunho exploratório de caráter qualitativo (GERHADT; SILVEIRA, 2009) e um método comparativo mediante a prática de um desenvolvimento de software através de uma plataforma *Low-Code*. Diante da temática abordada, o método da revisão será dedutivo, partindo da análise de artigos científicos publicados nos últimos cinco anos em língua inglesa e portuguesa, visando alcançar a conclusão junto ao experimento prático de desenvolvimento de forma a trazer discussões e alcançar o que se objetiva a pesquisa (GIL, 2010).

A criação de uma aplicação através de uma plataforma *low-code* com objetivo de demonstrar suas facilidades para construção ocorre partindo da plataforma *Salesforce*, utilizando o criador de aplicativos *Lightning* para criar um exemplo simplificado de um CRM, onde teremos uma tela de início (Figura 3) que traz informações importantes como compromissos e tarefas do dia; *Leads* que são os clientes em potencial; Contas que representam as empresas cliente; Contatos que são as pessoas que trabalham nas contas (empresas); as Oportunidades que são os negócios em andamento e por fim, um *Chatter* para possibilitar a interação entre os colaboradores.

Figura 3 – Tela de criação do aplicativo *Lightning*

Detalhes e marca do aplicativo

Dê ao seu aplicativo Lightning um nome e uma descrição. Carregue uma imagem e escolha a cor de destaque para a sua barra de navegação.

Detalhes do aplicativo	Marca de aplicativo
<p>* Nome do aplicativo ⓘ</p> <input type="text" value="Dê um nome ao seu aplicativo..."/>	<p>Imagem ⓘ</p> <div> Carregar</div>
<p>* Nome do desenvolvedor ⓘ</p> <input type="text" value="Insira um nome de desenvolvedor..."/>	
<p>Descrição ⓘ</p> <input type="text" value="Insira uma descrição..."/>	<p>Valor hexadecimal de cor primária ⓘ</p> <div><input type="color" value="#0070D2"/> #0070D2</div>
<p>Opções de tema da organização</p> <input type="checkbox"/> Usar a imagem e a cor do aplicativo em vez do tema personalizado da organização	

Avançar

Fonte: Elaborado pelo autor, 2023.

4 RESULTADOS E DISCUSSÕES

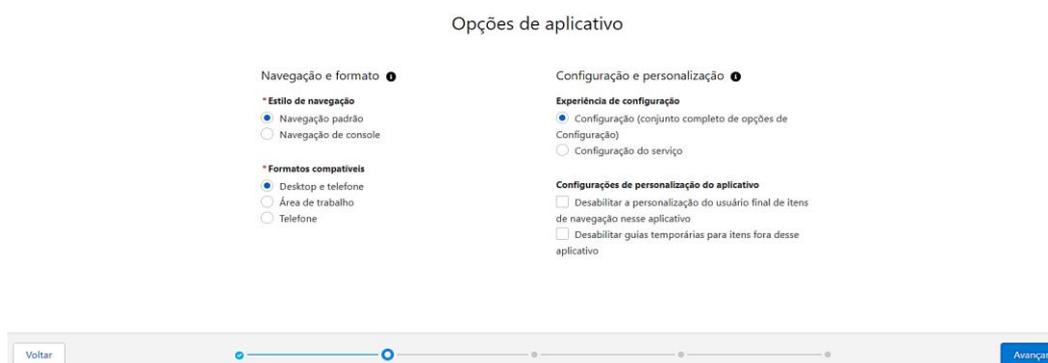
4.1 Criação da aplicação usando uma plataforma *Low-Code*

Para poder mostrar como funciona o desenvolvimento numa plataforma *Low-code*, vamos ver uma demonstração usando a plataforma da *Salesforce* que é a maior empresa de CRM (*Customer Relationship Management*) do mundo. Utilizando o

criador de aplicativos Lightning da Salesforce para criar um exemplo simplificado de um CRM, onde teremos: uma **Tela de Início**, que traz informações importantes como compromissos e tarefas do dia; termos os **Leads** que são os clientes em potencial; teremos as **Contas** que representam as empresas clientes; os **Contatos** que são as pessoas que trabalham nas contas (empresas); as **Oportunidades** que são os negócios em andamento; e um **Chatter** para possibilitar a interação entre os colaboradores.

Na tela de criação (Figura 3) é a primeira fase da criação do aplicativo, onde são informados o nome do aplicativo e o nome do desenvolvedor como campos de preenchimentos obrigatórios, além de descrição, uma imagem e seleção de cor para a página do aplicativo. Na segunda fase de criação do aplicativo (Figura 4), onde selecionamos o estilo de navegação, a navegação padrão traz itens em uma barra na parte superior da página, já a navegação de console mostra itens em uma barra lateral e abre os itens selecionados em uma nova guia. Podemos também permitir que os usuários finais não possam personalizar os itens para o aplicativo, mas vamos deixar o padrão sugerido pela ferramenta.

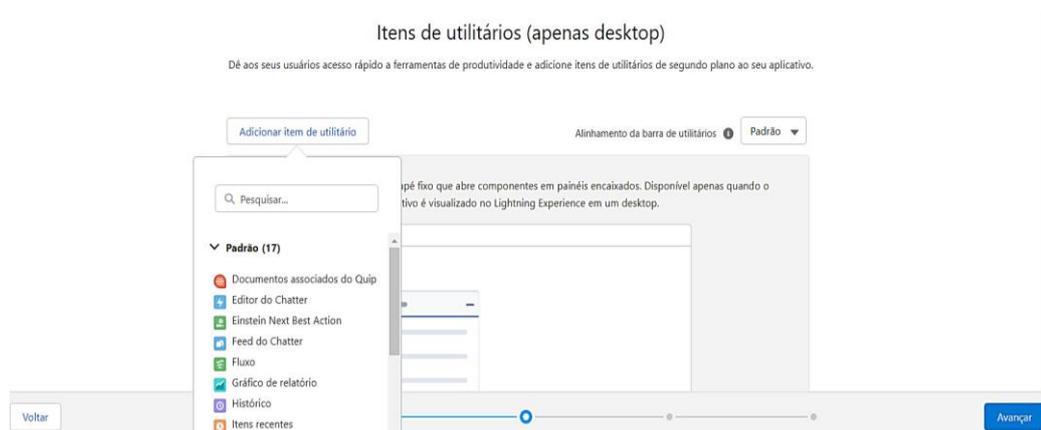
Figura 4 – Opções do aplicativo



Fonte: Elaborado pelo autor, 2023.

Na terceira fase da criação do aplicativo, onde é possível adicionar itens utilitários (Figura 5), já presentes na plataforma e que ajudam o usuário a aumentar sua produtividade. Não foi adicionado nenhum comando desses, apenas clicou-se em “Avançar”.

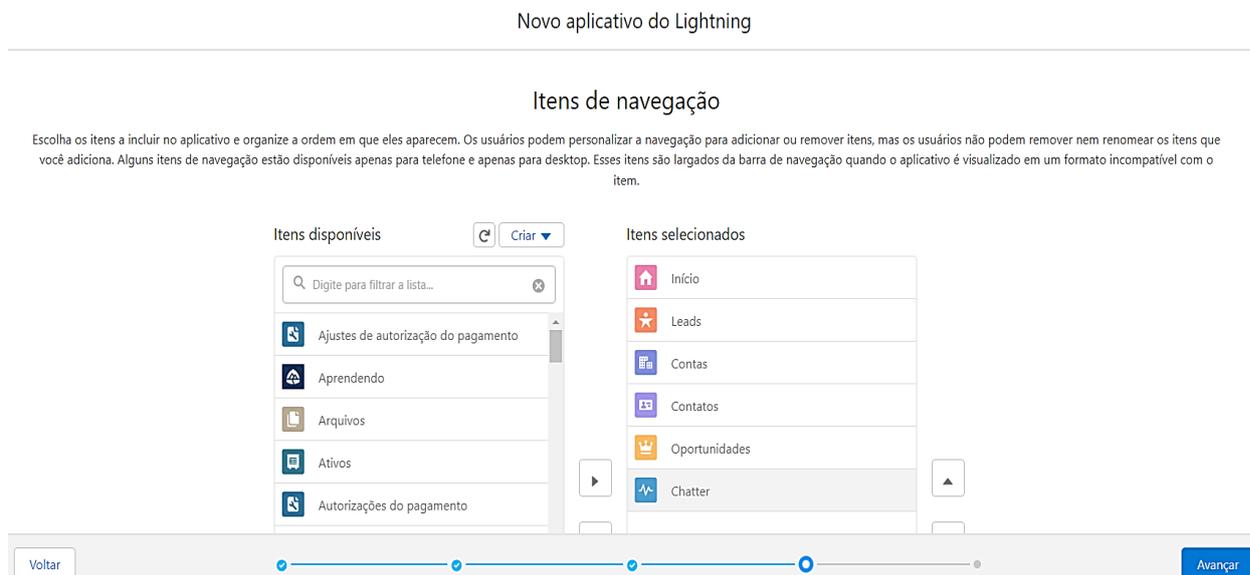
Figura 5 – Utilitários



Fonte: Elaborado pelo autor, 2023.

Na etapa seguinte (Figura 6), selecionamos os itens presentes na nossa aplicação, vamos adicionar os itens que mencionamos na descrição do aplicativo, que são: Início, Leads, Contas, Contatos, Oportunidades e Chatter.

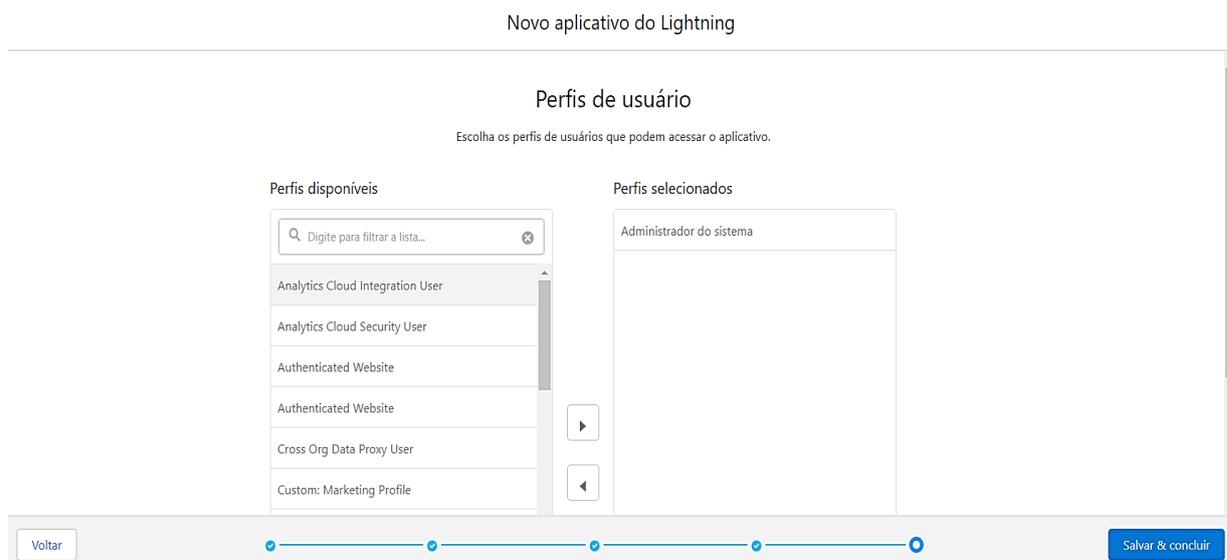
Figura 6 – Itens de navegação



Fonte: Elaborado pelo autor, 2023.

Nesta que é a última fase da criação do nosso aplicativo, selecionamos os perfis de usuário (Figura 7) que possuem acesso a nossa aplicação, no nosso caso foi selecionado apenas o perfil “Administrador do sistema”, que possui acesso a todos os recursos da plataforma.

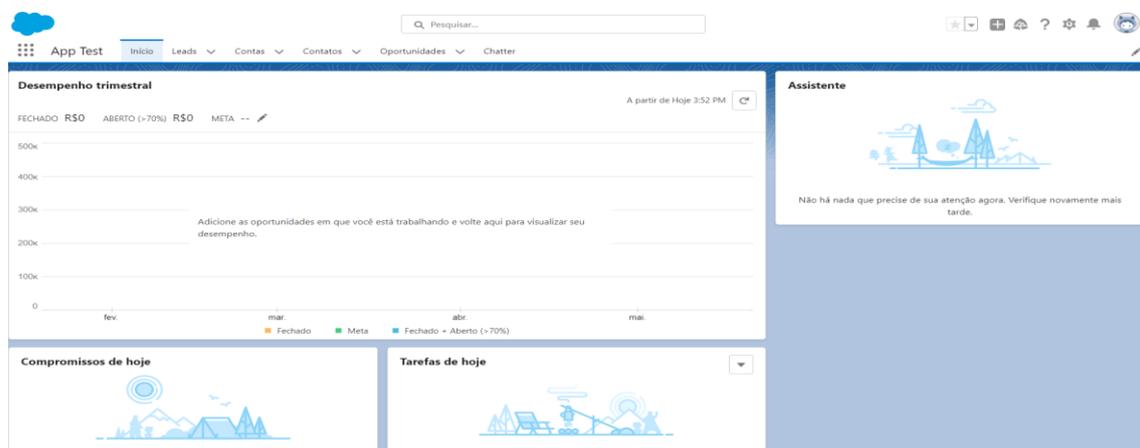
Figura 7 – Perfis de usuário



Fonte: Elaborado pelo autor, 2023.

Ao finalizar as etapas de criação do nosso aplicativo, esse é o resultado da nossa aplicação (Figuras 8-13). Na tela de início temos informações como o desempenho trimestral, compromissos do dia, tarefas entre outros, que podem ser ajustados de acordo com as necessidades do usuário.

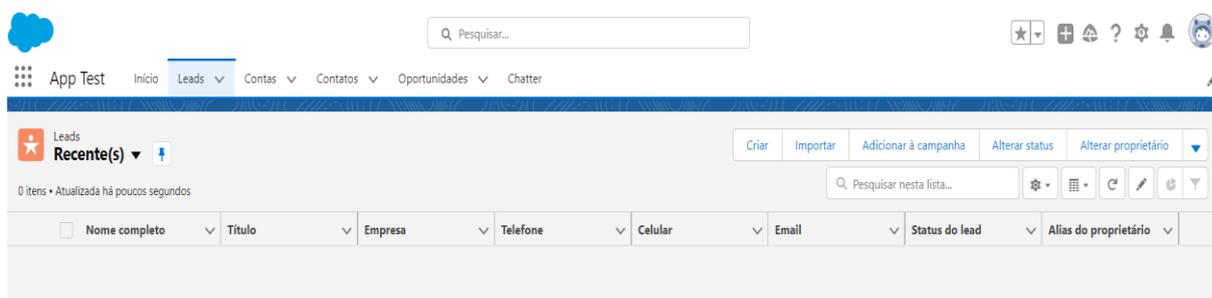
Figura 8 – Tela de início



Fonte: Elaborado pelo autor, 2023.

Essa tela (Figura 9) exibe uma lista com os *leads* presentes na nossa organização, como não foi povoada, exibe uma lista vazia. Podemos criar um novo *lead* ou importá-lo.

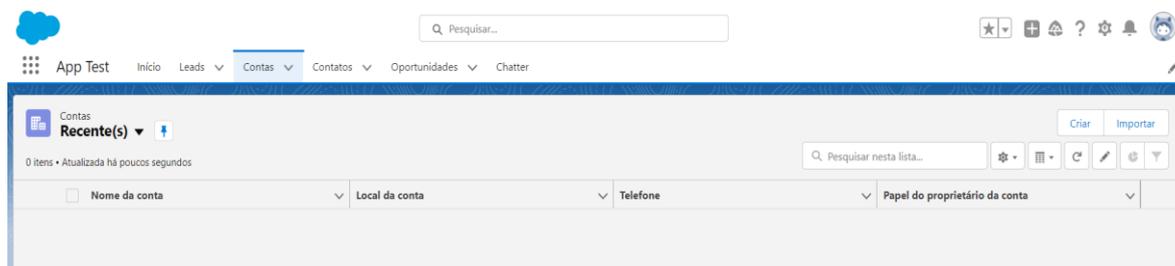
Figura 9 – Leads



Fonte: Elaborado pelo autor, 2023.

Esta tela (Figura 10) exibe uma lista com as contas presentes na nossa organização.

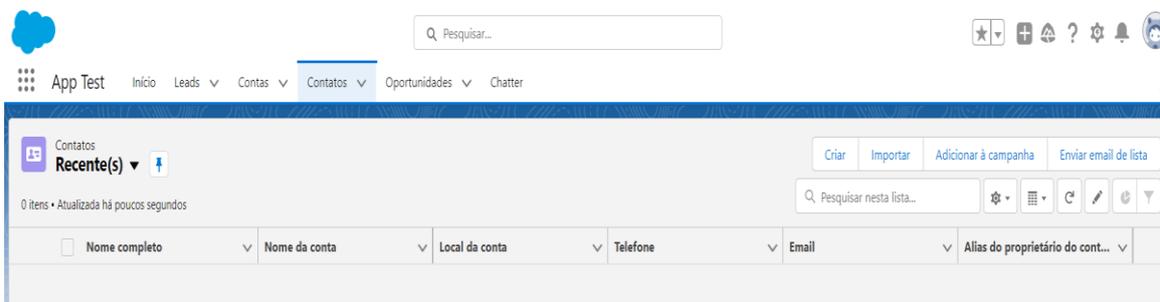
Figura 10 – Contas



Fonte: Elaborado pelo autor, 2023.

Essa tela (Figura 11) exibe os contatos presentes na nossa organização.

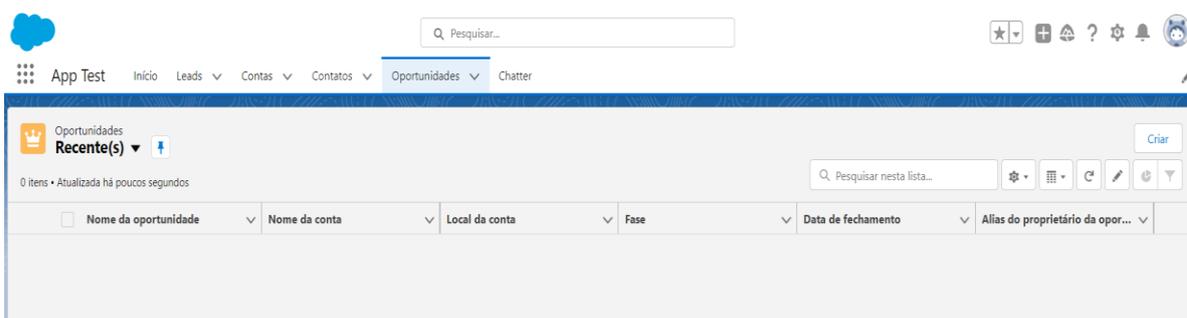
Figura 11 – Contatos



Fonte: Elaborado pelo autor, 2023.

Essa tela (Figura 12) exibe as oportunidades presentes na nossa organização.

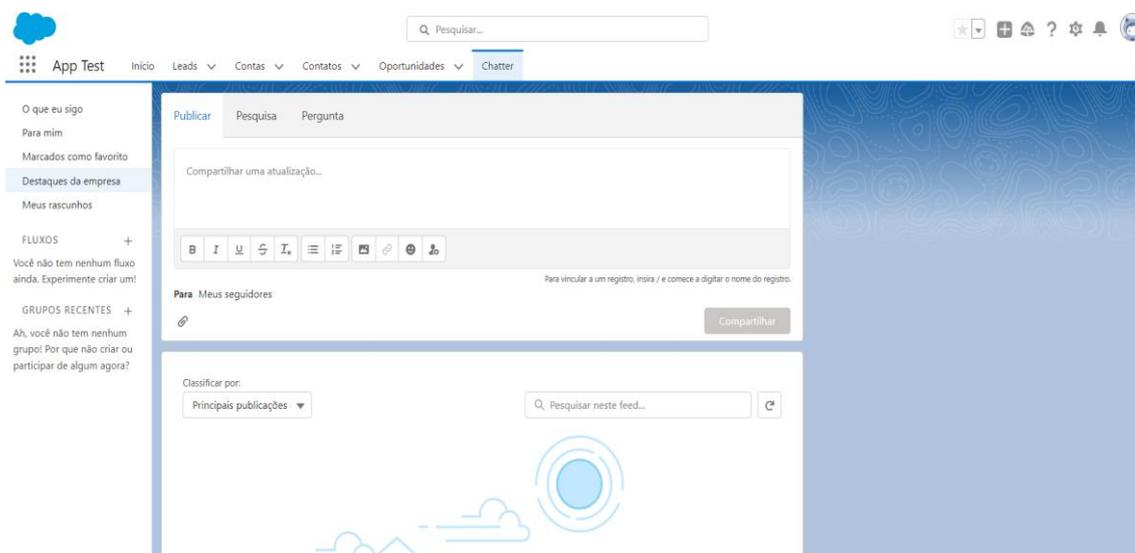
Figura 12 – Oportunidade



Fonte: Elaborado pelo autor, 2023.

Por fim temos o *Chatter* (Figura 13), onde podemos interagir com as pessoas da organização, publicando mensagens, perguntas ou pesquisa.

Figura 13 - Chatter



Fonte: Elaborado pelo autor, 2023.

Ainda nesta etapa, é possível anexar imagens, mencionar pessoas que fazem parte da organização entre outras funcionalidades. Ao finalizarmos a demonstração considerando que nosso aplicativo possui um escopo bem simples, podemos concluir que a plataforma é de fácil uso e entendimento, podemos criar objetos personalizados sem a necessidade de codificação e incluí-los na nossa aplicação sem a necessidade de outra aplicação para gerenciar os dados, pois a própria plataforma cuida disso.

4.2 Relação entre a programação prática e a literatura analisada

4.2.1 Vantagens do uso do low-code

Uma pesquisa realizada por Luo et al. (2021) utilizando o *Stack OverFlow* e *Reddit* com o intuito de responder à questão: “Quais são os benefícios do desenvolvimento *Low-code*?”, identificou 210 posts que discutiam sobre os benefícios do desenvolvimento *low-code*, onde foram identificados os 16 principais benefícios na perspectiva dos praticantes e que apresentou os seguintes resultados:

Quadro 1 – Benefícios do desenvolvimento *low-code*

BENEFÍCIOS	EXEMPLO	QUANT.
Desenvolvimento mais rápido	Trazer o app mais rápido ao mercado	65
Fácil uso e aprendizagem	Curva de aprendizado baixa	52
Custos mais baixos	Executar o aplicativo de código aberto de graça	18
Unidades prontas para uso	Módulos de fácil instalação	14
Amigável para novatos	Ajuda na criação de software de modo fácil	10
Melhor qualidade do sistema	Produz app de alta qualidade, seguro, escalável e de fácil manutenção	7
Forte capacidade de integração e expansão	Integração ao Drools e ao Drools fusion, permitindo modelar e executar cenários de negócios complexos	7
Esforço mínimo	Poupar tempo sem escrever códigos	7
Melhor personalização	Personalizar e dimensionar sem esforço atendendo as necessidades do cliente	6
Intuição perceptiva	Boa responsividade fora da caixa	5
Usabilidade superior	Completamente flexível e capaz de fazer quase tudo	5
Melhor experiência ao usuário	Fornecer ferramenta gráfica de fácil uso para transformar o design, os dados, criar mensagens, variáveis, condições, etc.	4
Implantação fácil	Para dispositivos móveis (Android e IOS)	4
Custo-benefício	Modo econômico de desenvolver app	3
Governança de TI aprimorada	Não exige confiabilidade total dos especialistas em negócios nos desenvolvedores	2
Melhor desenvolvimento da equipe	Boa adaptação ao desenvolvimento em equipe	1

Fonte: Adaptado de Luo et al. (2021, p. 8).

Outros autores discutem os benefícios do desenvolvimento *Low-code* podemos destacar: *Low-code* está quebrando barreiras técnicas para trabalhar no espaço de automação, facilitando a colaboração suave entre negócios e desenvolvimento (FAURA, 2021). Como a parte principal do código já está desenvolvida, os usuários só precisam configurar os aplicativos em vez de codificá-los manualmente (SANCHIS et al., 2020). Para Hirzel (2022) as plataformas *low-code* tendem a ser fáceis de ler, seja utilizando blocos que já são familiares para o especialista de domínio ou usando blocos intuitivos.

Experiência de usuário: As plataformas *low-code* oferecem ferramentas visuais de design de página para os usuários. Diferentes temas podem ser criados e usados para cada usuário (ŞAHINASLAN(a); SAHINASLAN(b); SABANCIOGLU, 2021). Manutenção: Plataformas *Low-code* são constantemente atualizadas por seus desenvolvedores, o que significa que uma plataforma deve ser construída de forma que seja fácil de manter (CRUZ, 2021).

Flexível e modular: Como o software permite que os usuários criem outros softwares facilmente, a plataforma deve ser intuitiva. No entanto, deve suportar casos de uso avançados e isso só é possível se a plataforma fornecer um grau de flexibilidade aos usuários (CRUZ, 2021). Os desenvolvedores podem criar facilmente seu próprio processo de gerenciamento de fluxo de trabalho e fluxo de autorização como eles desejam (ŞAHINASLAN(a); SAHINASLAN(b); SABANCIOGLU, 2021).

Base de dados: Os campos, componentes e tipos de dados desejados para serem usados no aplicativo são apresentados na plataforma, dispensando uma aplicação separada, para manutenção e gerenciamento dos dados (ŞAHINASLAN(a); SAHINASLAN(b); SABANCIOGLU, 2021). Escalável: Essas plataformas são desenvolvidas a partir de uma perspectiva global e possuem recursos mais escaláveis (ŞAHINASLAN(a); SAHINASLAN(b); SABANCIOGLU, 2021).

Privacidade: como os aplicativos podem ser desenvolvidos por usuários sem profundo conhecimento técnico, as empresas confiam em seus funcionários e essas tarefas de desenvolvimento geralmente não são terceirizadas, mas realizadas internamente, o que aumenta a confidencialidade (SANCHIS et al., 2020).

4.2.2 Desvantagens do uso do low-code

A mesma pesquisa realizada por Luo et al. (2021), buscava identificar as limitações da abordagem *low-code* e usando a mesma abordagem para a coleta dos dados, foram identificadas 15 limitações (Quadro 2) na perspectiva dos praticantes, os resultados foram os seguintes:

Quadro 2 – Desvantagens do desenvolvimento *low-code*

LIMITAÇÕES E DESAFIOS	EXEMPLO	QUANT.
Alta curva de aprendizagem	Necessário aprender mais sobre a ferramenta para trabalhar nela	21
Preço alto	Cobram por usuário, compra de pacotes de 50 ou 100 usuários	13
Falta de customização	Restrições no design e layouts	11
Carregamento e publicação lentos	Velocidades baixas	9
Menos poderoso que programação	Menos funcional que a linguagem de programação completa	6
Alta complexidade	Complicados de usar	6
Necessidade de codificação	Em certos níveis a depender da complexidade do problema é necessário uso de código ou criação personalizada	5
Sem acesso ao código-fonte	Não é possível pegar o código-fonte e usar em outro lugar	4
Não é fácil de usar	Apesar de ser ótimo, não é fácil de usar em comparação a escolher um app que já está escrito	4
Limitação a desenvolvedores experientes	A maioria das ferramentas são projetadas como prototipagem tornando o desenvolvimento difícil para inexperientes	4

Bloqueio de fornecedor	Quando desenvolvem, eles hospedam, em caso de aumento nos valores, o tempo inativo ou em reconstrução trará dados perdidos	3
Dificuldade na manutenção e depuração	O suporte contínuo e manutenção da plataforma é um risco adicional	3
Dificuldade na integração	Problema para fazer a interface do usuário, em armazenar dados e cálculos para trabalhar juntos	3
Experiência de usuário hostil	Apresenta inclinações íngremes e por vezes hostis ao usuário UX	2
Necessidade de conhecimento básico em programação básica	Requer código em algum momento do desenvolvimento	2

Fonte: Adaptado de Luo et al. (2021, p. 8).

Outros autores apontam alguns riscos, por exemplo, Richardson (2016) fala que pequenos vendedores vendem soluções sem o devido gerenciamento tecnológico e clientes possuem pouco entendimento sobre como as plataformas *low-code* se encaixam no seu portfólio. As ferramentas *low-code* são uma faca de dois gumes. Elas facilitam o trabalho em equipe, mas problemas com segurança e governança de dados podem não ser uma prioridade para não desenvolvedores (FAURA, 2021).

Um fator contra essas tecnologias são os custos de licenciamento, que normalmente são mais altos em comparação com tecnologias baseadas em código Trigo (2022). Para Hirzel (2022) a principal fraqueza do *low-code* é que as plataformas não são sempre autoexplicativas.

5 CONSIDERAÇÕES FINAIS

Com base na pesquisa é possível evidenciar que os autores consideram que a abordagem *low-code* vem crescendo e ocupando mais espaço no mercado, pois traz agilidade no desenvolvimento de software e permite que não-programadores possam criar suas aplicações, porém existem questionamentos sobre suas limitações, que envolvem a dependência de modelos pré-estabelecidos pela plataforma, a falta de habilidade técnica para usar as plataformas e dificuldades para personalizar as aplicações.

Os autores Waszkowski, Sanchis et al., Chen, Hirzel, Trigo e Şahinaslan et al. discutem a inserção de não-programadores no mundo do desenvolvimento *low-code* mas a falta de conhecimento técnico pode trazer problemas na definição do escopo dos projetos e por mais generalista que seja a plataforma nem tudo pode ser feito através da abordagem *low-code*, principalmente com relação a personalização, por mais simples e atrativa que seja uma plataforma *low-code*, muitas das necessidades dos usuários finais ainda são atendidas através da programação tradicional, fazendo com que sua adoção não implique na não necessidade de programadores mas podendo sim ser uma forma de acelerar o processo de desenvolvimento de softwares.

Waszkowski diz que o *low-code* permite que o programador gaste menos tempo pensando na sintaxe do código e coloque mais ênfase em projetar a estética e a funcionalidade da aplicação. O custo é considerado um entrave na adoção dessas plataformas, em sua maioria o preço está relacionado com o número de usuários finais e suas licenças de uso possuem valor elevado.

REFERÊNCIAS

- CHEN, Whai-En; LIN, Yi-Bing; YEN, Tai-Hsiang; PENG, Syuan-Ru; LIN, Yun-Wei. Device Talk: A No-Code Low-Code IoT Device Code Generation. **Sensors**, v. 22, n. 4942, p. 1-19, 2022. Disponível em: <https://www.mdpi.com/1424-8220/22/13/4942>. Acesso em: 10 dez. 2022.
- CRUZ, Mauro A. A. da; PAULA, Heitor T. L. de; CAPUTO, Bruno P. G.; MAFRA, Samuel B.; LORENZ, Pascal; RODRIGUES, Joel J. P. C. OLP – A RESTful Open Low-Code Plataform. **Future Internet**, v. 13, n. 249, p. 1-14, sep. 2021. Disponível em: <https://www.mdpi.com/1999-5903/13/10/249>. Acesso em: 10 set. 2022.
- ELSHAN, Edona; DICKHAUT, Ernestine; EBEL, Philipp. An Investigation of Why Low Code Platforms Provide Answers and New Challenges. **Hawaii International Conference on System Sciences (HICSS)**, Maui, Hawaii, p. 1-12, 2023. Disponível em: <https://iwi.unisg.ch/publications/an-investigation-of-why-low-code-platforms-provide-answers-and-new-challenges/>. Acesso em: 12 fev. 2023.
- FAURA, Miguel Valdes. Low-Code and DevOps: Friends or Foes? Disponível em: <https://devops.com/low-code-and-devops-friends-or-foes>. Acesso em: 21 fev. 2023.
- GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. 2. reimpr. São Paulo: Atlas, 2010. 184 p. il.
- GOOGLE CLOUD. AppSheet: desenvolvimento de apps sem código. **Google Cloud**. Disponível em: <https://cloud.google.com/appsheet?hl=pt-br#section-5>. Acesso em: 16 fev. 2023.
- HIRZEL, Martin. Low-Code Programming Models. **Cornell University**, USA, v. 1, p. 1-10, may, 2022. Disponível em: <https://arxiv.org/abs/2205.02282>. Acesso em: 10 jan. 2023.
- IHIRWE, Felicien; DI RUSCIO, Davide; MAZZINI, Silvia; PIERINI, Pierluigi; PIERANTONIO, Alfonso. Low-code Engineering for Internet of things: A state of research. **MODELS '20 Companion**, Virtual Event, Canada, p. 1-8, oct. 2020. Disponível em: <https://arxiv.org/abs/2009.01876>. Acesso em: 15 out. 2022.
- KUMARVIVEK. O que é o Power Apps? - **Power Apps**. Disponível em: <https://learn.microsoft.com/pt-br/power-apps/powerapps-overview>. Acesso em: 16 fev. 2023.
- LAKATOS Eva Maria. **Metodologia científica**. 6. ed. 4. reimpr. São Paulo: Atlas, 2011. 314 p. ISBN 9788522466252.
- LUO, Yajing; LIANG, Peng; WANG, Chong; SHAHIN, Mojtaba; ZHAN, Jing. Characteristics and Challenges of Low-Code Development: The Practitioners' Perspective. **Cornell University**, USA, v. 11, n. 15, p. 1-11, oct. 2021. Disponível em: <https://arxiv.org/abs/2107.07482>. Acesso em: 10 out. 2022.

MARCONI, Marina de Andrade. **Fundamentos de metodologia científica**. 7. ed. 2. reimpr. São Paulo: Atlas, 2010. 297 p. il.

MENDIX. Collaborative Visual Development. **Mendix**. Disponível em: <https://www.mendix.com/collaborative-visual-development>. Acesso em: 16 fev. 2023.

MUNDO OUTSYSTEMS. O que é OutSystems Platform. 2017. **Mundo OutSystems**. Disponível em: <https://mundoutsystems.com/o-que-e-outsistemas>. Acesso em: 16 fev. 2023.

OUTSYSTEMS. The State of Application Development. Is IT Ready for Disruption? **OutSystems**, Boston, MA, USA, 2019. Disponível em: <https://www.outsystems.com/local-gov/-/media/053D5BCC32364C2993C8D0BAFA880DB1.ashx>. Acesso em: 11 nov. 2022.

RICHARDSON, Clay; RYMER, John R. Vendor Landscape: the fractured, fertile terrain of low-code application platforms. **FORRESTER**, v. 15, p. 1-23, 2016. Disponível em: <https://www.forrester.com/report/Vendor-Landscape-The-Fractured-Fertile-Terrain-Of-LowCode-Application-Platforms/RES122549>. Acesso em: 12 fev. 2023.

RIMOL, Meghan. Gartner Says the Majority of Technology Products and Services Will Be Built by Professionals Outside of IT by 2024. **Gartner**. 14 June 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: 16 fev. 2023.

SAHAY, Apurvanand; INDAMUTSA, Arsene; DI RUSCIO, Davide; PIERANTONIO, Alfonso. Supporting the understanding and comparison of low-code development platforms. In: **46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**, Portoroz, Slovenia, p. 171-178, 2020. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9226356/authors>. Acesso em: 01 fev. 2023.

ŞAHINASLAN(a), Ender; SAHINASLAN(b), Onder; SABANCIOGLU, Mehmet. Low-Code Application Platform in Meeting Increasing Software Demands Quickly: SetXRM. **AIP Publishing**, v. 2334, e070007, p. 1-5, mar. 2021. Disponível em: <https://aip.scitation.org/doi/10.1063/5.0042213>. Acesso em: 10 jan. 2023.

SALESFORCE. Lightning App Builder. **Salesforce**. Disponível em: https://help.salesforce.com/s/articleView?id=sf.lightning_app_builder_overview.htm&type=5. Acesso em: 16 fev. 2023.

SANCHIS, Raquel; GARCÍA-PERALES, Oscar; FRAILE, Francisco; POLER, Raul. Low-Code as Enabler of Digital Transformation in Manufacturing Industry. **Appl. Sci.**, v. 10, n. 12, p. 1-17, 2020. Disponível em: <https://www.mdpi.com/2076-3417/10/1/12>. Acesso em: 09 out. 2022.

SILVA, Igor Carneiro da. **Possibilitando a manipulação de comportamentos em produtos virtuais sem conhecimento de tecnologia**. 2021. 51 f. Monografia

(Graduação em Engenharia da Computação) – Universidade Federal de Pernambuco, Recife. 2021. Disponível em: https://www.cin.ufpe.br/~tg/2020-2/TG_EC/tg_ics.pdf. Acesso em: 15 out. 2022.

SOUZA, Jhonata Lima de. **A contribuição do Low Code no âmbito educacional: um mapeamento sistemático da literature.** 2022. 65 f. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) - Universidade Federal de Pernambuco, Recife. 2022. Disponível em: https://www.cin.ufpe.br/~tg/2021-2/tg_SI/TG_jls3.pdf. Acesso em: 12 out. 2022.

TRIGO, António; VARAJÃO, João; ALMEIDA, Miguel. Low-Code Versus Code-Based Software Development: which wins the productivity game? **IEEEExplore IT Professional**, Portugal, v. 24, n. 5, p. 61-68, sept. /oct. 2022. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9967415>. Acesso em: 10 set. 2022.

WASZKOWSKI, Robert. Low-code platform for automating business processes in manufacturing. **IFAC-PapersOnLine**, v. 52, n. 10, p. 376-381, 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405896319309152>. Acesso em: 15 set. 2022.

AGRADECIMENTOS

À Deus, por toda força e sabedoria concedida para que fosse possível junto a minha persistência, concluir esta graduação.

Aos meus pais e a minha irmã, que sempre foram minha ponte de apoio, por entenderem os momentos de ausência e por suportarem os dias difíceis.

Aos meus colegas de turma, que pude compartilhar momentos inesquecíveis durante toda a graduação.

Aos meus amigos, por todas as palavras de incentivo e momentos de risos e apoio moral.

A coordenação do curso do Curso de Ciências da Computação, por seu empenho.

Ao professor e orientador, Edson Holanda Cavalcante Júnior, pela paciência e educação ao longo de toda a jornada acadêmica bem como desta orientação.

Aos professores da Banca Examinadora, por suas contribuições e correções para o aperfeiçoamento deste trabalho.

A todos os professores que fazem parte do curso de Computação, pelos ensinamentos transmitidos.