

**AS CARACTERÍSTICAS DAS FERRAMENTAS LOW-CODE E NO-CODE E SEUS  
IMPACTOS NO MERCADO ATUAL**  
***THE CHARACTERISTICS OF LOW-CODE AND NO-CODE TOOLS AND THEIR  
IMPACTS ON THE CURRENT MARKET***

Felipe de Souza Máximo Rodrigues – felipesouza.fs449@gmail.com  
Faculdade de Tecnologia de Taquaritinga (Fatec) – Taquaritinga – SP – Brasil

Giuliano Scombatti Pinto – giuliano.pinto@fatec.sp.gov.br  
Faculdade de Tecnologia de Taquaritinga (Fatec) – Taquaritinga – SP – Brasil

DOI: 10.31510/infa.v22i1.2245

Data de submissão: 10/04/2025

Data do aceite: 26/06/2025

Data da publicação: 30/06/2025

**RESUMO**

As ferramentas *Low-code* oferecem uma solução intermediária entre flexibilidade e facilidade de uso, acelerando o processo de desenvolvimento, enquanto as ferramentas *No-code* democratizam o desenvolvimento, tornando-o acessível a um público mais amplo sem a necessidade de codificação manual. O objetivo do presente trabalho é explorar o impacto das ferramentas *Low-code* e *No-code* no desenvolvimento de software, abordando suas definições, analisando suas características, semelhanças e diferenças, além de explorar como influenciam o mercado e a formação de profissionais. A metodologia utilizada é a pesquisa bibliográfica. Os resultados possibilitaram identificar a importância dessas inovações na rápida entrega de soluções tecnológicas e na transformação do panorama educacional para profissionais de TI. Ambas as abordagens têm vantagens, como a aceleração do desenvolvimento e a democratização do processo, mas também apresentam desafios, como a limitação de complexidade. Concluiu-se que, apesar dos desafios, as duas ferramentas abordadas são catalisadoras de mudanças significativas, moldando o presente e o futuro do desenvolvimento de software, e devem ser atentamente acompanhadas por aqueles que buscam eficiência e adaptabilidade na indústria de tecnologia.

**Palavras-chave:** Desenvolvimento. Ferramentas *Low-code*. Ferramentas *No-code*. Inovação.

**ABSTRACT**

Low-code tools offer an intermediary solution between flexibility and ease of use, speeding up the development process, while No-code tools democratize development, making it accessible to a broader audience without the need for manual coding. The aim of this study is to explore

the impact of Low-code and No-code tools on software development, addressing their definitions, analyzing their characteristics, similarities, and differences, as well as exploring how they influence the market and professional formation. The methodology used is bibliographic research. The results made it possible to identify the importance of these innovations in the rapid delivery of technological solutions and in the transformation of the educational landscape for IT professionals. Both approaches have advantages, such as speeding up development and democratizing the process, but they also present challenges, such as complexity limitations. It was concluded that, despite the challenges, the two tools discussed are catalysts for significant change, shaping the present and future of software development, and should be closely monitored by those seeking efficiency and adaptability in the technology industry.

**Keywords:** Development. Low-code Tools. No-code Tools. Innovation. Keywords.

## 1. INTRODUÇÃO

Nos últimos anos, a demanda por soluções de software tem crescido exponencialmente em diversos setores da economia, impulsionando a necessidade de desenvolvimento de aplicativos de forma rápida e eficiente. No entanto, o desenvolvimento de software tradicional, que envolve a codificação manual de cada aspecto de um aplicativo, é um processo complexo e demorado, muitas vezes exigindo habilidades técnicas avançadas e recursos consideráveis.

Nesse contexto, surgiram as ferramentas *Low-code* e *No-code* como alternativas inovadoras para simplificar e acelerar o desenvolvimento de software. Essas ferramentas visam atender a uma ampla gama de usuários, desde desenvolvedores profissionais até pessoas sem conhecimento técnico avançado, permitindo que eles criem aplicativos e softwares personalizados de maneira mais acessível e rápida.

Segundo Masili (2023), a importância das tecnologias *Low-code* e *No-code* é cada vez maior devido à falta de conhecimento interno das empresas sobre linguagens de programação tradicionais e à dificuldade em recrutar programadores profissionais para acelerar o processo de desenvolvimento das aplicações.

O objetivo deste trabalho é explorar e analisar as ferramentas *Low-code* e *No-code*, suas características, funcionalidades e impacto no desenvolvimento de software. Será examinado como essas ferramentas estão revolucionando a forma como aplicativos são criados, democratizando o acesso à criação de software e acelerando o ciclo de desenvolvimento.

## 2. FERRAMENTAS LOW-CODE E NO-CODE

### 2.1 Ferramentas Low-code

#### 2.1.1 Definição

As ferramentas *Low-code* são plataformas de desenvolvimento de software que permitem a criação de aplicativos com um nível reduzido de codificação manual (Rice, 2023). Elas são projetadas para simplificar o processo de desenvolvimento, tornando-o mais acessível a uma variedade de profissionais, desde desenvolvedores com experiência limitada até desenvolvedores experientes.

Segundo Salnik (2022), *Low-code* é uma metodologia de desenvolvimento de software na qual o desenvolvedor pode criar aplicativos com pouca codificação. Dito isso, estas ferramentas de automação são ambientes de desenvolvimento integrado (IDEs) orientados visualmente, projetados para o desenvolvimento rápido de aplicativos.

#### 2.1.2 Principais Características

Segundo Rice (2023) e SAP(s.d), as principais características das ferramentas *Low-code* incluem:

- **Interface Gráfica:** As ferramentas *Low-code* oferecem uma interface de usuário gráfica e intuitiva, que permite aos desenvolvedores e não desenvolvedores criar aplicativos arrastando e soltando elementos visuais, como botões, campos de entrada e tabelas.
- **Componentes Reutilizáveis:** Elas fornecem uma biblioteca de componentes reutilizáveis, como módulos de autenticação, integrações com APIs e funcionalidades de banco de dados, economizando tempo e esforço na criação de funcionalidades comuns.
- **Automatização de Tarefas:** As ferramentas *Low-code* incluem automações que simplificam fluxos de trabalho complexos. Isso permite que os desenvolvedores configurem ações automatizadas em resposta a eventos específicos, reduzindo a necessidade de codificação manual de lógica de negócios.
- **Integração de Dados:** Elas facilitam a integração com sistemas externos e fontes de dados, tornando possível a criação de aplicativos que coletam, processam e apresentam informações de várias fontes.

- Escalabilidade: Embora as ferramentas *Low-code* ofereçam rapidez e facilidade no desenvolvimento inicial, muitas delas também suportam personalizações e extensões avançadas, permitindo que aplicativos cresçam com as necessidades da organização.

Bratincevic e Koplowitz (2021) destacam que as ferramentas *Low-code* oferecem uma gama de recursos, como criação de formulários, automação de fluxos de trabalho, geração de relatórios e integração de dados, permitindo que usuários desenvolvam aplicativos completos com rapidez e facilidade.

### 2.1.3 Exemplos de ferramentas Low-code

A seguir serão apresentadas algumas ferramentas e recursos que podem ser utilizados:

- WordPress: principal plataforma usada atualmente no desenvolvimento de blogs, sites institucionais e portfólios. Sua interface inclui as principais funções necessárias para criar e administrar um site de boa qualidade, mas também possui recursos para programadores criarem seus próprios temas e plugins ou modificar o de terceiros.
- Mendix: uma plataforma de desenvolvimento de aplicativos hospedada em nuvem, com desenvolvimento orientado a modelos. A programação é intuitiva e, uma vez concluída, pode ser executada de qualquer máquina, sem necessidade de uma base de código.
- OutSystem: uma plataforma ideal para quem precisa criar apps para IOS ou Android, mas tem pouco ou nenhum conhecimento de linguagem de programação. A interface segue um sistema “*drag and drop*”.

## 2.2 Ferramentas No-code

### 2.2.1 Definição

As ferramentas *No-code* são plataformas de desenvolvimento de software que possibilitam a criação de aplicativos sem a necessidade de codificação manual. Elas são voltadas para um público ainda mais amplo do que as ferramentas *Low-code*, incluindo profissionais não técnicos, como gestores de negócios e designers (Masili, 2023).

Nakanishi (s.d) define que *No-code* refere-se a um ecossistema de ferramentas e metodologias que permitem que pessoas sem habilidades ou conhecimentos técnicos criem

software sem codificação. Isso ocorre porque as ferramentas sem código usam uma interface cheia de funcionalidades como arrastar e soltar, modelos prontos e componentes pré-construídos que permitem criar software visualmente.

### 2.2.2 Principais Características

As principais características das ferramentas *No-code* destacadas por Oluwaleye (2023) e Rice (2023) incluem:

- **Interface de Montagem Visual:** As ferramentas *No-code* oferecem uma interface de montagem visual altamente intuitiva, onde os usuários podem construir aplicativos combinando blocos de lógica, elementos de interface e funcionalidades.
- **Ausência de Codificação:** A característica mais marcante é a ausência total ou quase total de codificação. Os usuários podem criar aplicativos simplesmente escolhendo opções e configurando regras, sem a necessidade de escrever código.
- **Velocidade de Desenvolvimento:** As ferramentas *No-code* são conhecidas por sua rapidez no desenvolvimento. Os aplicativos podem ser prototipados e implantados em questão de horas ou dias, em vez de semanas ou meses.
- **Acessibilidade:** Essas ferramentas tornam o desenvolvimento de software acessível a um público mais amplo, permitindo que pessoas sem formação técnica criem aplicativos que atendam às suas necessidades específicas.
- **Integração Simples:** Assim como as ferramentas *Low-code*, as ferramentas *No-code* também facilitam a integração de dados e sistemas externos, permitindo que os aplicativos acessem informações de várias fontes.

Além disso, as ferramentas *No-code* destacam-se por possuírem um editor WYSIWYG (“*what you see is what you get*”, em português, “o que você vê é o que você obtém”), ou seja, o desenvolvedor pode ver como será o projeto que está desenvolvendo em tempo real e qualquer alteração é refletida imediatamente. (SAP, s.d)

### 2.1.3 Exemplos de ferramentas No-code

A seguir serão apresentadas algumas ferramentas e recursos que podem ser utilizados:

- Webflow: plataforma na qual é possível criar sites com design de alto nível e estrutura profissional usando o sistema “*drag and drop*”, não sendo necessário que o desenvolvedor tenha algum conhecimento de programação.
- Airtable: plataforma que simplifica a gestão de projetos, define prazos e monitora orçamentos. Sua interface gráfica economiza tempo e a plataforma se integra com muitas outras ferramentas. O Airtable também oferece um banco de dados flexível que pode ser personalizado de acordo com as necessidades do projeto.
- Bubble: plataforma que permite desenvolver web apps inovadores sem precisar de HTML ou CSS. Possui interface intuitiva na qual é possível utilizar a função “*drag and drop*”, modelos e plug-ins disponíveis no mercado para construir um aplicativo. Além disso, a plataforma oferece muitas integrações com outras ferramentas.

### 3. METODOLOGIA

Para realizar esta pesquisa, empregou-se uma abordagem abrangente que incluiu revisão bibliográfica e avaliação de tendências de mercado. A revisão bibliográfica foi conduzida para compreender os conceitos fundamentais de ferramentas *Low-code* e *No-code*, analisando trabalhos acadêmicos, livros, websites e artigos relacionados.

## 4 RESULTADOS

### 4.1 Semelhanças

Em relação a facilidade de uso, tanto as ferramentas *Low-code* quanto as *No-code* são conhecidas por sua simplicidade de utilização. Elas adotam interfaces gráficas intuitivas que permitem aos usuários criar aplicativos arrastando e soltando elementos visuais, o que reduz a necessidade de escrever código manualmente.

No que se refere a rapidez no desenvolvimento, ambas as abordagens são projetadas para acelerar o desenvolvimento de software. Elas permitem a criação de protótipos e aplicativos funcionais em um tempo significativamente menor do que o desenvolvimento tradicional (Rice, 2023).

No que tange a integração de dados, tanto as ferramentas *Low-code* quanto as *No-code* facilitam a integração com sistemas externos, APIs e fontes de dados. Isso permite que os aplicativos acessem informações de várias fontes e realizem tarefas de automação.

Por fim, em termos de flexibilidade, embora as ferramentas *No-code* sejam mais restritas em termos de flexibilidade, ambas as abordagens oferecem a capacidade de personalização e extensão dos aplicativos. Os usuários podem adaptar os aplicativos para atender às necessidades específicas de seus negócios.

Segundo Jednaszewski (2024), as plataformas *Low-code* e *No-code* fornecem os meios para construir aplicativos sem escrever código. Com uma abordagem visual, os desenvolvedores não precisam entender vários tipos de linguagens de programação.

## 4.2 Diferenças

A diferença principal entre plataformas de desenvolvimento de baixo código e sem código está na quantidade de conhecimento em programação que o usuário precisa. Plataformas de desenvolvimento de *Low-code* (LCDPs) exigem habilidades básicas em codificação para que os usuários possam criar e integrar aplicativos complexos, enquanto plataformas de desenvolvimento *No-code* (NCDPs) não exigem nenhum conhecimento em programação (SAP, s.d).

Em relação ao público-alvo, as ferramentas *Low-code* são mais voltadas para desenvolvedores e profissionais de TI que desejam acelerar o desenvolvimento de aplicativos e simplificar tarefas rotineiras. Por outro lado, as ferramentas *No-code* visam um público mais amplo, incluindo gestores de negócios, designers e qualquer pessoa que queira criar aplicativos sem conhecimento técnico profundo (Johannessen; Davenport, 2021).

No que tange a complexidade de aplicativos, as ferramentas *Low-code* são mais adequadas para o desenvolvimento de aplicativos complexos que requerem lógica de negócios avançada e integrações profundas com sistemas existentes. As ferramentas *No-code* são mais adequadas para aplicativos simples e fluxos de trabalho que envolvem principalmente interações baseadas em formulários e automações simples (SAP, s.d).

*Low-code* concentra-se na criação de aplicativos e sistemas, não apenas em portais que podem ser conectados a outras soluções. *No-code*, por outro lado, concentra-se em soluções de front-end que não requerem personalização (Sydle, 2023).

Por fim, em relação à flexibilidade e controle, as ferramentas *Low-code* oferecem maior flexibilidade e controle sobre o desenvolvimento, permitindo que os desenvolvedores

personalizem a lógica do aplicativo e implementem funcionalidades avançadas. As ferramentas *No-code*, devido à sua natureza simplificada, podem ser mais limitadas em termos de personalização.

#### 4.3 Escolha da Abordagem Certa

Jednaszewski (2024) diz que para escolher uma plataforma, há mais do que apenas tecnologia. Como qualquer estratégia de desenvolvimento, deve-se planejar como alinhar os resultados com o que os usuários realmente desejam e precisam.

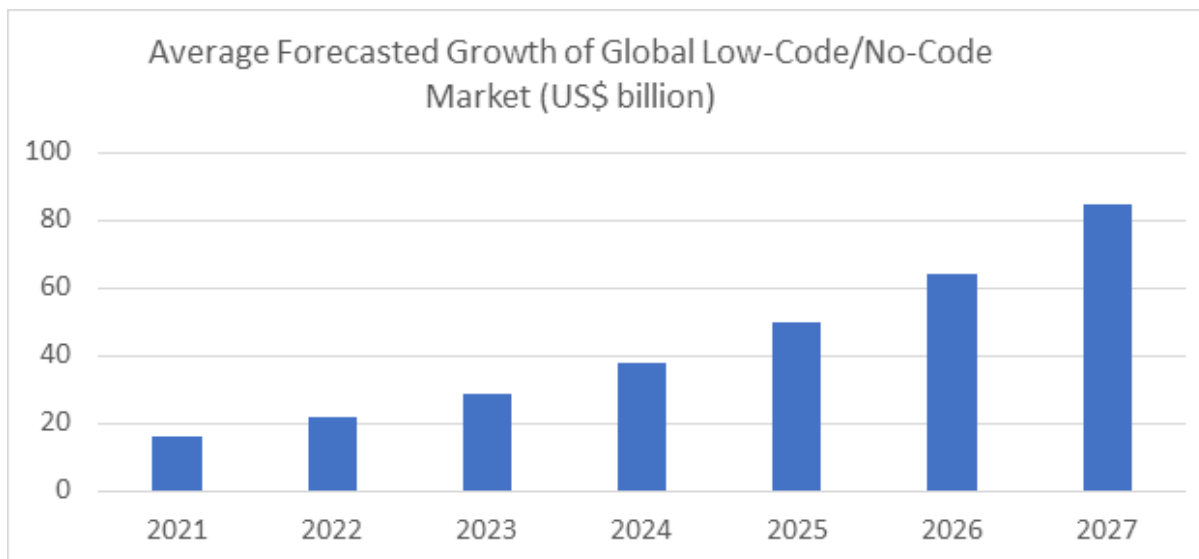
Sendo assim, conclui-se que a escolha entre ferramentas *Low-code* e *No-code* depende das necessidades específicas de um projeto e das habilidades da equipe envolvida. Para aplicativos simples e rápidos, as ferramentas *No-code* podem ser ideais. No entanto, para projetos mais complexos e personalizados, as ferramentas *Low-code* podem ser a melhor opção, permitindo que desenvolvedores experientes desempenhem um papel significativo. (SAP, s.d)

#### 4.4 Impacto no mercado

As ferramentas *Low-code* e *No-code* permitem que as empresas desenvolvam aplicativos de forma mais rápida, reduzindo o tempo de entrega de projetos e acelerando a resposta às demandas do mercado. Rice (2023) e Oluwaleye (2023) destacam que o desenvolvimento mais rápido e a redução da necessidade de desenvolvedores altamente especializados podem resultar em uma economia de custos significativa para as companhias que optem por utilizar tais ferramentas.

Além disso, Rice (2023) cita que essas ferramentas democratizam o desenvolvimento de aplicativos, permitindo que uma gama mais ampla de usuários contribua com soluções de software. Eles oferecem maior velocidade, flexibilidade e colaboração, ajudando as organizações a impulsionar a inovação e a atingir seus objetivos de transformação digital de forma mais eficaz.

A Figura 1 mostra o crescimento médio previsto das ferramentas *Low-code* e *No-code* até o ano de 2027 em bilhões de dólares.

**Figura 1 - Crescimento médio previsto das ferramentas *Low-code* e *No-code***

Fonte: SpreadsheetWeb (2021)

Entretanto, há também alguns desafios. As ferramentas *No-code* podem não ser adequadas para aplicativos altamente complexos ou personalizações avançadas, o que limita seu uso em certos cenários (SAP. s.d). Além disso, as organizações podem ficar dependentes dos fornecedores das ferramentas, o que pode gerar preocupações sobre o controle e a segurança dos dados.

Apesar de serem mais acessíveis, essas ferramentas ainda requerem algum treinamento para uso eficaz, especialmente para aproveitar todas as funcionalidades disponíveis.

#### **4.5 Impacto na formação de novos profissionais**

As ferramentas *No-code* tornam o desenvolvimento de software acessível a um público mais amplo, incluindo gestores de negócios, designers e profissionais de diversas áreas. Além disso, essas ferramentas podem servir como uma porta de entrada para o mundo da programação, permitindo que iniciantes ganhem confiança e experiência antes de se aprofundarem em codificação manual.

Em contrapartida, dependendo do uso excessivo de ferramentas *No-code*, os profissionais podem não desenvolver habilidades de codificação manual, o que pode limitar suas opções de carreira em desenvolvimento de software. A ausência de conhecimento em programação pode resultar em uma compreensão deficiente dos fundamentos de

desenvolvimento de software, o que pode prejudicar a solução de problemas e a criação de aplicativos mais complexos.

#### 4.6 Perspectivas futuras

O uso de ferramentas *Low-code* e *No-code* continuará a crescer à medida que a demanda por desenvolvimento de software rápido e acessível aumenta. Espera-se que essas ferramentas evoluam para abordar desafios como a escalabilidade de aplicativos e a personalização avançada, tornando-as ainda mais versáteis.

Para a formação de novos profissionais, é provável que as instituições de ensino e programas de treinamento incorporem o ensino de ferramentas *Low-code* e *No-code* como parte integrante do currículo, a fim de preparar os estudantes para um mercado de trabalho em constante evolução.

### 5. CONCLUSÃO

Neste trabalho, foram exploradas as ferramentas *Low-code* e *No-code* e a crescente influência delas no mundo do desenvolvimento de software. Ao longo do artigo, foram discutidas as definições, funcionalidades, semelhanças e diferenças, bem como seus impactos no mercado e na formação de novos profissionais.

Ficou evidente que essas abordagens estão revolucionando a maneira como aplicativos são criados, oferecendo uma série de vantagens, como aceleração do desenvolvimento, redução de custos e democratização do processo de criação de software. As ferramentas *Low-code*, com um equilíbrio entre flexibilidade e facilidade de uso, atendem a uma ampla gama de necessidades, enquanto as ferramentas *No-code* abrem as portas do desenvolvimento de software para um público ainda mais amplo, eliminando quase totalmente a necessidade de codificação manual.

No entanto, essas vantagens não estão isentas de desafios. As limitações de complexidade, a dependência de fornecedores e a necessidade de treinamento são questões que precisam ser consideradas. Além disso, a formação de novos profissionais no campo do desenvolvimento de software também está passando por transformações, à medida que as instituições de ensino e programas de treinamento se adaptam para incorporar o ensino dessas ferramentas.

À medida que a humanidade avança para o futuro, é importante reconhecer que as ferramentas *Low-code* e *No-code* continuarão a evoluir, enfrentando e superando esses desafios.

Elas têm o potencial de capacitar uma gama ainda mais ampla de pessoas a criar aplicativos personalizados, impulsionando a inovação e a eficiência em todos os setores.

Em resumo, as ferramentas *Low-code* e *No-code* representam um marco na democratização do desenvolvimento de software, promovendo a acessibilidade, a agilidade e a eficiência. Elas moldam o presente e o futuro da tecnologia, oferecendo novas perspectivas para o mercado e para a formação de profissionais, e devem ser acompanhadas de perto por aqueles que buscam se manter relevantes e competitivos na indústria de tecnologia da informação.

## REFERÊNCIAS

ADAM. **How Big is the Global Low-Code/No-Code Market and How Fast is it Growing?** 12 jul. 2021. Disponível em: <https://spreadsheetweb.com/how-big-is-the-global-low-code-no-code-market-and-how-fast-is-it-growing/>. Acesso em 30 mar. 2025.

BRATINCEVIC, John; KOPLOWITZ, Rob. **Low-Code Development Platforms For Professional Developers, Q2 2021.** 11 mai 2021. The Forrester Wave™:

BROCODERS COMPANY. **The Pros and Cons of Low-Code Development.** 6 mar. 2021. Disponível em: <https://hackernoon.com/the-pros-and-cons-of-low-code-development-4y2p33g9>. Acesso em 10 mar. 2025.

GLOBALFY. **3 melhores ferramentas no-code para expandir seu negócio em 2023.** 18 abr. 2023. Disponível em: <https://globalfy.com/blog/3-melhores-ferramentas-no-code/>. Acesso em 30 mar. 2025.

JEDNASZEWSKI, Marek. **No-Code vs. Low-Code Development: Differences, Similarities, and Use Cases.** Disponível em: <https://www.mendix.com/blog/understand-no-code-vs-low-code-development-tools/#no-code-vs-low-code-development-the-similarities>. Acesso em 30 mar. 2025.

JOHANNESSEN, Chris; DAVENPORT, Tom. **When Low-Code/No-Code Development Works — and When It Doesn't.** 22 jun. 2021. Disponível em: <https://hbr.org/2021/06/when-low-code-no-code-development-works-and-when-it-doesnt>. Acesso em: 8 mar. 2025.

MASILI, Giorgia. (2023). **No-Code Development Platforms: Breaking The Boundaries Between It And Business Experts.** 10 abr. 2023. International Journal of Economic Behavior, p.33-49, vol. 13, 2023.

NAKANISHI, Duncan Griffiths. **What is No-code?** Disponível em: <https://www.nocode.tech/article/what-is-no-code>. Acesso em 10 mar. 2025.

OLUWALEYE, James. **Low-Code and No-Code Development Tools.** 7 jun. 2023. Disponível em: <https://semaphoreci.com/blog/low-code-no-code-development-tools>. Acesso em: 6 mar. 2025.

REDAÇÃO CRONAPP. **Confira alguns exemplos de low-code!** 27 mar. 2023. Disponível em: <https://blog.cronapp.io/confira-alguns-exemplos-de-low-code/>. Acesso em 30 mar. 2025.

RICE, Abigail. **How low-code, no-code tools are changing the game.** 11 out 2023. Disponível em: <https://www.kentico.com/discover/blog/how-low-code-no-code-is-changing-the-game>. Acesso em 30 mar. 2025.

SALNIK, Rodion. **Low-Code Development: Why It Is Important and Why It Can't Replace Traditional Approach.** Disponível em: <https://brocoders.com/blog/low-code-development-approach>. Acesso em 30 mar. 2025.

SAP. **What is low-code/no-code application development?** Disponível em: <https://www.sap.com/products/technology-platform/low-code/what-is-low-code-no-code.html>. Acesso em 9 mar. 2025.

SYDLE. **Low Code: Why Is It Essential for Any Business?** 29 jun. 2023. Disponível em: <https://www.sydle.com/blog/low-code-what-is-essential-6148d1a8e83193407d19ff7a>. Acesso em: 6 mar. 2025.