

Universidade de Brasília – UnB  
Faculdade de Ciências e Tecnologias em Engenharia –  
FCTE  
Engenharia de Software

***Low-code* no apoio à Transformação  
Digital: uma abordagem de Design de  
Serviços**

Autores: João Pedro de Camargo Vaz e Vinícius  
Assumpção de Araújo

Orientadora: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Rejane M. C. Figueiredo

Brasília, DF

2025



João Pedro de Camargo Vaz e Vinícius Assumpção de Araújo

## ***Low-code* no apoio à Transformação Digital: uma abordagem de Design de Serviços**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade de Ciências e Tecnologias em Engenharia – FCTE

Orientador: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Rejane M. C. Figueiredo

Brasília, DF

2025

---

João Pedro de Camargo Vaz e Vinícius Assumpção de Araújo

*Low-code* no apoio à Transformação Digital: uma abordagem de Design de Serviços/ João Pedro de Camargo Vaz e Vinícius Assumpção de Araújo. – Brasília, DF, 2025- [122](#) p. : il. (algumas color.) ; 30 cm.

Orientador: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Rejane M. C. Figueiredo

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB

Faculdade de Ciências e Tecnologias em Engenharia – FCTE , 2025.

1. Low-Code. 2. Transformação Digital. 3. Design de serviços. 4. Inovação Tecnológica. 5. Eficiência Organizacional. I. Prof<sup>ª</sup>. Dr<sup>ª</sup>. Rejane M. C. Figueiredo. II. Universidade de Brasília. III. Faculdade de Ciências e Tecnologias em Engenharia – FCTE. IV. *Low-code* no apoio à Transformação Digital: uma abordagem de Design de Serviços

CDU

---

João Pedro de Camargo Vaz e Vinícius Assumpção de Araújo

## ***Low-code* no apoio à Transformação Digital: uma abordagem de Design de Serviços**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 11 de fevereiro de 2025:

---

**Prof. Dr<sup>a</sup>. Rejane Maria da Costa  
Figueiredo**  
Orientadora

---

**Prof. Dr. John Lenon Cardoso  
Gardenghi**  
Convidado 1

---

**Dr. Luiz Carlos Midayara Ribeiro  
Junior**  
Convidado 2

Brasília, DF  
2025

# Resumo

A transformação digital (TD) tem revolucionado as organizações ao promover inovação, eficiência e adaptação a um mercado dinâmico. No apoio à TD, de um lado, a abordagem de Design de Serviço (DS) desempenha um papel importante na criação de experiências de valor para os usuários, e de outro, abordagens e tecnologias que promovam o desenvolvimento, como o Desenvolvimento pelo Usuário Final (EUD), em que usuários finais, muitas vezes sem formação em programação, participam ativamente do desenvolvimento de software. Nesse contexto, tecnologias *low-code* emergem como catalisadoras, permitindo o desenvolvimento ágil de soluções com menor dependência de programação tradicional, enquanto o design de serviços colabora na criação de experiências centradas no usuário. Este trabalho investiga o quanto o ambiente de desenvolvimento *App Engine Studio* (APS) da *ServiceNow* apoia a transformação digital, tendo como objeto uma organização brasileira de economia mista que atua no mercado financeiro. Por meio de um estudo de caso, analisam-se a aceleração de desenvolvimento, a mitigação do *Shadow IT* e a integração de princípios do design de serviços. Os resultados destacam a capacidade do *low-code* de reduzir custos e prazos, fomentar a inovação e alinhar soluções tecnológicas às estratégias organizacionais. Como trabalhos futuros, expandir a análise para outras áreas e setores, reforçando o papel do *low-code* como alicerce da transformação digital.

**Palavras-chave:** Low-Code, Transformação Digital, Design de Serviços, Inovação Tecnológica, Eficiência Organizacional.

# Abstract

Digital Transformation (DT) has been revolutionizing organizations by fostering innovation, efficiency, and adaptation to a dynamic market. Supporting DT, on one hand, the Service Design (SD) approach plays a crucial role in creating valuable experiences for users, while on the other hand, approaches and technologies that promote development, such as End-User Development (EUD), empower end-users—often without programming skills—to actively participate in software development. In this context, *low-code* technologies emerge as catalysts, enabling the agile development of solutions with reduced dependence on traditional programming, while service design contributes to the creation of user-centered experiences. This study investigates the extent to which the App Engine Studio (APS) development environment by ServiceNow supports DT, focusing on a Brazilian mixed-economy organization operating in the financial market. Through a case study, it examines the acceleration of development, the mitigation of Shadow IT, and the integration of SD principles. The results highlight *low-code*'s ability to reduce costs and timelines, foster innovation, and align technological solutions with organizational strategies. As future work, it is suggested to expand the analysis to other areas and sectors, reinforcing the role of *low-code* as a cornerstone of digital transformation.

**Keywords:** Low-Code, Digital Transformation, Service Design, Technological Innovation, Organizational Efficiency.

# Lista de ilustrações

Figura 1 – Principais razões para se usar plataformas <i>low-code</i> (Adaptado)(SANCHIS et al., 2019). . . . .	37
Figura 2 – Processo de <i>Design de Serviço</i> na digitalização de serviços <i>low-code</i> (Adaptado)(IRIARTE et al., 2023). . . . .	42
Figura 3 – Domínios dos estudos no que tange a abordagem <i>low-code</i> (BUCAIONI; CICCETTI; CICCIOZZI, 2022). . . . .	46
Figura 4 – Força da plataforma <i>ServiceNow</i> e sua capacidade de impactar a transformação digital juntamente com o design de serviços (BRATINCEVIC; KOPLOWITZ, 2021) . . . . .	57
Figura 5 – Quadrante de líderes - <i>Atualização do quadrante “mágico” de empresas líderes de mercado</i> (MATVITSKY; DAVIS; JAIN, 2024) . . . . .	58
Figura 6 – Cronograma do Projeto de Pesquisa . . . . .	67
Figura 7 – Processo de criação de aplicações usando o APS (ServiceNow, 2024a) . . . . .	75
Figura 8 – Processo de criação de aplicações usando o APS (ServiceNow, 2024a) . . . . .	75

# Lista de tabelas

Tabela 1 – Comparação entre Digitização e Digitalização (CORESYSTEMS, 2024). . . . .	22
Tabela 2 – Algumas das <i>features</i> suportadas por plataformas LCNC, adaptado de (ROKIS; KIRIKOVA, 2023) . . . . .	34
Tabela 3 – Protocolo de Estudo de Caso (BRERETON et al., 2008) . . . . .	61
Tabela 4 – Impacto do APS na Organização e Métricas Numéricas . . . . .	98



# Lista de abreviaturas e siglas

APS	App Engine Studio
CSCW	trabalho cooperativo suportado por computador
DS	Design de Serviços
EUD	<i>End User Development</i>
HCD	<i>Human Centered Design</i>
HCI	interação humano-computador
LCNC	<i>Low-code/no-code</i>
P&D	Pesquisa e Desenvolvimento
SES	<i>Sprint Execution Score</i>
SESAT	<i>Sprint Execution Satisfaction</i>
UnB	Universidade de Brasília

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>1.1</b>	<b>Considerações Iniciais do Capítulo</b>	<b>12</b>
<b>1.2</b>	<b>Contexto</b>	<b>12</b>
<b>1.3</b>	<b>Questão de Pesquisa</b>	<b>15</b>
<b>1.4</b>	<b>Objetivos</b>	<b>16</b>
1.4.1	Objetivo Geral	16
1.4.2	Objetivos Específicos	16
<b>1.5</b>	<b>Metodologia</b>	<b>17</b>
1.5.1	Plano Metodológico	17
<b>1.6</b>	<b>Organização do Trabalho</b>	<b>18</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>19</b>
<b>2.1</b>	<b>Considerações Iniciais do Capítulo</b>	<b>19</b>
<b>2.2</b>	<b>Transformação Digital</b>	<b>19</b>
2.2.1	Digitização e Digitalização no contexto da TD	21
<b>2.3</b>	<b>Design de Serviços</b>	<b>23</b>
<b>2.4</b>	<b>Desenvolvimento pelo Usuário Final - EUD</b>	<b>26</b>
<b>2.5</b>	<b><i>Shadow IT</i></b>	<b>29</b>
<b>2.6</b>	<b><i>Abordagem Low e No-Code</i></b>	<b>30</b>
2.6.1	Plataformas de desenvolvimento <i>Low e No-Code</i>	32
2.6.2	Benefícios do <i>low-code/no-code</i>	35
2.6.3	Comparação entre <i>low-code e no-code</i>	37
2.6.4	Princípios do desenvolvimento <i>Low Code</i>	39
2.6.5	<i>DS</i> no desenvolvimento <i>low-code</i>	41
2.6.6	Ciclo de vida	43
2.6.7	Áreas de atuação	45
2.6.8	Mercado <i>low-code/no-code</i>	47
<b>2.7</b>	<b>Considerações Finais do Capítulo</b>	<b>59</b>

<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>60</b>
<b>3.1</b>	<b>Considerações Iniciais da Seção</b>	<b>60</b>
<b>3.2</b>	<b>Planejamento de Pesquisa</b>	<b>60</b>
<b>3.3</b>	<b>Coleta de Dados</b>	<b>61</b>
<b>3.4</b>	<b>Análise</b>	<b>62</b>
<b>3.5</b>	<b>Relatório</b>	<b>62</b>
<b>3.6</b>	<b>Atividades do Protocolo de Estudo de Caso</b>	<b>62</b>
3.6.1	<i>Background</i>	62
3.6.2	<i>Design</i>	63
3.6.3	Seleção de Caso	63
3.6.4	Procedimentos e papéis do estudo de caso	64
3.6.5	Coleta de Dados	65
3.6.6	Análise	65
3.6.7	Validade do plano	66
3.6.8	Limitações do Estudo	66
3.6.9	Relatório	66
3.6.10	Cronograma	66
3.6.11	Apêndices	66
<b>3.7</b>	<b>Considerações Finais do Seção</b>	<b>67</b>
<b>4</b>	<b>PLATAFORMA SERVICENOW</b>	<b>68</b>
<b>4.1</b>	<b>Considerações Iniciais do Capítulo</b>	<b>68</b>
<b>4.2</b>	<b>Visão Geral da <i>ServiceNow</i></b>	<b>68</b>
<b>4.3</b>	<b><i>App Engine Studio</i></b>	<b>69</b>
4.3.1	Papéis no Desenvolvimento com <i>App Engine Studio</i>	70
4.3.2	Princípios da Metodologia Ágil no <i>APS</i>	73
4.3.3	Processo de Desenvolvimento do <i>APS</i>	74
<b>4.4</b>	<b>Considerações Finais do Capítulo</b>	<b>77</b>
<b>5</b>	<b>IMPACTO DO <i>LOW-CODE</i> NA ORGANIZAÇÃO</b>	<b>79</b>
<b>5.1</b>	<b>Considerações Iniciais do Capítulo</b>	<b>79</b>
<b>5.2</b>	<b>Procedimentos de Coleta de Dados</b>	<b>79</b>
5.2.1	Entrevistas com Representantes da Organização	80

5.2.1.1	<i>Shadow IT</i> e o contexto do <i>low-code</i> na <i>Organização</i> . . . . .	82
5.2.1.2	Perspectiva Estratégica da plataforma <i>low-code</i> no contexto de TD . . .	83
5.2.1.3	TD impulsionada pelo <i>low-code</i> . . . . .	85
5.2.1.4	Aspectos de Desenvolvimento de Software com <i>low-code</i> . . . . .	87
5.2.1.5	Princípios de Design de Serviços para a promoção da TD . . . . .	90
5.2.2	Análise Documental . . . . .	92
5.2.2.1	Padrões e Tendências Identificados . . . . .	94
5.2.2.2	Divergências e Explicações: . . . . .	96
5.2.2.3	Discussão Ampliada dos Resultados . . . . .	97
<b>5.3</b>	<b>Validação dos Resultados</b> . . . . .	<b>99</b>
5.3.1	Limitações Documentadas . . . . .	100
5.3.2	Restrições Metodológicas . . . . .	101
5.3.3	Impactos nas Recomendações Finais . . . . .	101
<b>5.4</b>	<b>Considerações Finais</b> . . . . .	<b>101</b>
<b>6</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b> . . . . .	<b>103</b>
<b>6.1</b>	<b>Principais Conclusões</b> . . . . .	<b>103</b>
<b>6.2</b>	<b>Contribuições para a Prática Organizacional</b> . . . . .	<b>104</b>
<b>6.3</b>	<b>Limitações e Trabalhos Futuros</b> . . . . .	<b>105</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>106</b>
	<b>APÊNDICE A – CARTA DE APRESENTAÇÃO</b> . . . . .	<b>112</b>
	<b>APÊNDICE B – ROTEIRO DAS ENTREVISTAS</b> . . . . .	<b>114</b>
	<b>APÊNDICE C – CARTA DE AGRADECIMENTO</b> . . . . .	<b>121</b>

# 1 Introdução

## 1.1 Considerações Iniciais do Capítulo

Neste Capítulo apresenta-se o contexto deste trabalho, com uma breve revisão bibliográfica, o apontamento da questão de pesquisa, o objetivo geral e os específicos, bem como a metodologia para o alcance desses objetivos. Por fim, apresenta-se a estrutura desta monografia.

## 1.2 Contexto

A Transformação Digital (TD) mostra-se como uma necessidade para organizações que desejam se manter competitivas em um mercado em constante evolução. A digitalização e a digitização dos processos empresariais, a adoção de novas tecnologias e a reconfiguração dos modelos de negócios são elementos-chave dessa transformação ([VERINA; TITKO, 2019](#)). [Kagan \(2021\)](#) ressalta que as organizações que conseguem implementar com sucesso a Transformação Digital são capazes de melhorar significativamente sua eficiência operacional, sua capacidade de inovar e gerar savings — isto é, economias em custos e recursos decorrentes da otimização de processos e redução de redundâncias.

[Scupola e Mergel \(2022\)](#) definem TD como um esforço abrangente para revisar processos centrais das organizações, evoluindo de uma transição analógica para uma digital, envolvendo uma revisão completa de políticas, processos atuais e necessidades dos usuários, que resultam na criação de novos serviços digitais e na revisão dos serviços existentes. Esse processo holístico visa atender às necessidades dos usuários, desenvolver novas formas de entrega de serviços e possibilitar a expansão da base de usuários, ao mesmo tempo em que permite às organizações operar com maior eficiência e sustentabilidade financeira, maximizando savings.

Nesse cenário, a abordagem de *Design de Serviços* (DS) desempenha um papel importante, focando na criação de experiências de valor para os usuários

(clientes, cidadãos, funcionários), dado que um dos princípios da abordagem DS é a centralidade no humano. O Design de Serviços integra diversas metodologias e práticas para garantir que as soluções desenvolvidas atendam às reais necessidades dos usuários e criem valor de maneira sustentável (STICKDORN et al., 2022). A colaboração engloba diversos *stakeholders*, desde desenvolvedores, funcionários, parceiros, e fortemente os usuários finais, assegurando que todas as perspectivas dos *stakeholders* sejam consideradas no processo de design de uma solução. O DS consiste em uma abordagem que equilibra as necessidades do usuário final (clientes/cidadãos) e as exigências do negócio/organização, visando criar experiências do serviço contínuas e de alta qualidade. Essa abordagem permite que as organizações compreendam seus serviços do ponto de vista do cliente. O DS, por meio de métodos colaborativos que envolvam tanto clientes/cidadãos quanto membros internos e parceiros das organizações, possibilita as organizações obterem uma visão completa e verdadeira de seus serviços.

O DS se baseia também no *design thinking* para construção de soluções colaborativas e nos conceitos de Codesign e Cocriação. O conceito de Codesign se destaca como uma prática essencial no envolvimento de diversos participantes (*stakeholders*), como os tomadores de decisão, desenvolvedores, parceiros, fornecedores e demais colaboradores relevantes no processo do design. Nessa abordagem, todos os participantes colaboram ativamente para criar soluções tecnológicas, contribuindo com suas perspectivas e conhecimentos específicos. A colaboração multidisciplinar promove uma compreensão mais profunda dos problemas a serem resolvidos, facilita a identificação de oportunidades inovadoras e fortalece o compromisso dos envolvidos com a implementação das soluções. Possibilitando assim, a melhoria do processo de aceitação das soluções, assim como da eficácia e da relevância dos serviços resultantes, ao garantir que todas as vozes sejam ouvidas e consideradas ao longo do processo de desenvolvimento.

A Cocriação, por outro lado, se destaca pela participação do usuário final na geração de valor e inovação. Diferente dos métodos tradicionais de design, a cocriação pressupõe a participação ativa dos usuários finais (clientes/cidadãos) como integrantes do processo, explicitando suas necessidades e experiências durante o desenvolvimento ou validação de soluções. Assim, os usuários deixam de

ser sujeitos passivos e se tornam parceiros ativos, atuando desde a ideação até a implementação dos serviços. Ao democratizar o design e valorizar as vozes dos usuários finais, a cocriação facilita a resolução de problemas complexos e fomenta a inovação, condições essenciais para a transformação digital e a competitividade atuais (TRISCHLER; PERVAN; SCOTT, 2017; LEE; GRINEVICH; CHIPULU, 2023).

Ainda no cenário de Transformação Digital, uma das abordagens que alavancam a produção de serviços de Tecnologia da Informação (TI) é o Desenvolvimento pelo Usuário Final (em inglês, *End User Development* - EUD). O EUD está intimamente relacionado ao processo de Transformação Digital, dado que permite aos usuários finais, muitas vezes sem formação em programação, participarem ativamente do desenvolvimento de software. O EUD capacita esses usuários a criar e customizar software (aplicativos, sistemas) de acordo com suas necessidades específicas, aumentando a agilidade e a inovação nas organizações (LIEBERMAN; PATERNÒ; WULF, 2006). O EUD possibilita que as soluções tecnológicas estejam mais alinhadas com as necessidades reais dos usuários e promova inclusão e colaboração entre diferentes perfis de desenvolvedores e usuários finais (MACÍAS, 2012).

Todavia, no contexto de TD, de DS e de EUD, o fenômeno conhecido como *Shadow IT* também se mostra relevante, ao refletir a busca de agilidade e autonomia pelos usuários finais, com o desenvolvimento de tecnologias e ferramentas não aprovadas oficialmente pela governança organizacional, o que gera um desafio de governança. Nesse sentido, o desenvolvimento *low-code* surge como ferramenta que permite o controle e a inovação, alinhando com as diretrizes organizacionais e as necessidades de autonomia dos usuários. A gestão eficaz de *Shadow IT* torna-se, assim, um elemento crucial na criação de estratégias que integrem inovação descentralizada e governança estruturada, a exemplo de plataformas de desenvolvimento *low-code*.

Nesse sentido, de um ponto de vista mais técnico e operacional, as abordagens *Low* e *No-Code* complementam as práticas de DS e EUD ao serem disponibilizadas em plataformas que permitam o desenvolvimento rápido de software com pouco ou nenhum código. Essas plataformas buscam democratizar a inovação

tecnológica, permitindo que desenvolvedores cidadãos — indivíduos sem formação técnica — contribuam ativamente para o desenvolvimento de aplicações (ROKIS; KIRIKOVA, 2023).

Para Sanchis et al. (2019), a *era low-code* é apresentada como um catalisador nas organizações, facilitando a transformação digital e permitindo que essas se adaptem mais rapidamente às demandas do mercado e às necessidades dos clientes, promovendo assim uma cultura de inovação contínua. Para Kagan (2021), esse modelo possibilita não apenas agilidade, mas também uma inclusão maior na contribuição para inovações tecnológicas.

Neste contexto, este trabalho tem como *objeto de pesquisa* uma organização brasileira de economia mista que atua no mercado financeiro, denominada aqui ficticiamente de *Organização*, cuja estratégia de Transformação Digital está fundamentada em 03 (três) pilares: *Promoção da Experiência Digital e Inovação*; *Promoção da Integração dos Processos e das Soluções*; e *Promoção da Cultura Digital para a Inovação e para a Equidade*.

A *Organização* emprega o ambiente de desenvolvimento *App Engine Studio* (APS) da empresa *ServiceNow*, que segundo a literatura, possibilita a criação rápida e eficiente de aplicativos empresariais personalizados, a partir de diferentes papéis, com funções cruciais no desenvolvimento e na implantação de aplicativos (ServiceNow, 2024a), e pode promover a colaboração entre *desenvolvedores técnicos* e *desenvolvedores não técnicos* e a *cocriação* (com usuários finais) para garantir soluções alinhadas às suas necessidades e expectativas.

A empresa *ServiceNow* fornece uma plataforma de automação de fluxos de trabalho hospedados na nuvem e que possibilita às organizações implementarem seus processos, incluindo operações de TI, atendimento ao cliente, recursos humanos, entre outras (ServiceNow, 2024b).

### 1.3 Questão de Pesquisa

A crescente demanda por soluções de TI nas empresas tem mostrado-se um desafio significativo para muitas organizações, especialmente quando se busca



otimizar custos (*savings*) e mitigar problemas de *Shadow IT*. A necessidade de inovação constante, a pressão por tempos de resposta rápidos e a escassez de desenvolvedores qualificados criam um ambiente em que os métodos tradicionais de desenvolvimento de software frequentemente não conseguem acompanhar as exigências do mercado. Nesse sentido, o uso de plataformas de desenvolvimento *low-code* surge como uma alternativa capaz de proporcionar ganhos financeiros e operacionais, ao mesmo tempo em que reduz a ocorrência de *Shadow IT*, uma vez que amplia a governança de TI e promove uma maior colaboração entre áreas de negócio e equipes técnicas.

Contudo, a adoção de plataformas *low-code* por si só não garante uma TD eficaz. Estudos apontam que o DS assume papel relevante nesse processo, pois a abordagem de *cocriação* de valor com as soluções tecnológicas, inspirada pelo EUD, impulsiona tanto a redução de custos como o controle de aplicações externas aos padrões de TI. Dessa forma, as empresas contam com insumos fundamentais para a mitigação do *Shadow IT* e para a promoção da TD, ao mesmo tempo em que obtêm *savings* relevantes no ciclo de desenvolvimento.

Nesse contexto, a pergunta de pesquisa deste trabalho é: *Como as plataformas de desenvolvimento de software low-code, empregadas com a abordagem de design de serviços promovem a transformação digital?*

## 1.4 Objetivos

### 1.4.1 Objetivo Geral

O objetivo geral deste trabalho, tendo como cenário a *Organização*, é: *Investigar como a abordagem Low-code, utilizando o ambiente de desenvolvimento App Engine Studio da ServiceNow, empregada no contexto de Design de Serviços, tem apoiado a estratégia de TD da Organização.*

### 1.4.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

1. Investigar o quanto o *low-code* contribui para a mitigação do *Shadow IT* na *Organização*, e conseqüentemente, para a TD;
2. Investigar o emprego do *low-code* sob a perspectiva dos *pilares de TD* da *Organização*;
3. Analisar o quanto *low-code* tem acelerado o desenvolvimento de soluções digitais e de novas funcionalidades para a Integração de Processos e Soluções da *Organização*;
4. Investigar como os princípios do *Design de Serviços* foram integrados nas soluções desenvolvidas com a abordagem *low-code/no-code* na evolução do ambiente organizacional.

## 1.5 Metodologia

### 1.5.1 Plano Metodológico

Conforme Brereton et al (2008), um plano metodológico possui quatro fases principais: Planejamento; Coleta de Dados; Análise de Dados; e Relatórios, adotadas no plano metodológico deste trabalho.

Essas fases compreendem, de forma sucinta:

- **Planejamento da Pesquisa:** nessa fase apresenta-se o contexto do pesquisa com a revisão bibliográfica, pergunta de pesquisa, definição dos objetivos do trabalho e a definição de um plano metodológico, com as escolhas metodológicas;
- **Coleta dos Dados:** essa fase compreende o levantamento e a aplicação das técnicas de coleta de dados. Neste trabalho adotou-se a Revisão documental, Revisão bibliográfica e a técnica Estudo de Caso;
- **Análise dos Dados:** nessa fase realizam-se a interpretação e análise dos dados coletados, assim como a análise da validade do trabalho;

- **Relatório:** por fim, o relatório de pesquisa é constituído pelos resultados deste trabalho e materializado como esta monografia.

Neste plano metodológico, para o emprego da técnica Estudo de Caso, adotou-se o *Protocolo de Estudo de Caso* proposto por [Brereton et al. \(2008\)](#).

## 1.6 Organização do Trabalho

Este trabalho está organizado em 4 Capítulos. Este, Capítulo 1 - Introdução, apresenta a contextualização, o problema, objetivos de pesquisa, e a metodologia aplicada. Ademais, os restantes estão estruturados no seguinte formato:

Capítulo 2 - Referencial Teórico - apresenta-se a base teórica deste trabalho, com conceitos como: TD; DS; EUD; e a Abordagem *Low-Code*;

Capítulo 3 - Materiais e Métodos - Detalha o planejamento metodológico desta pesquisa, abrangendo as etapas de coleta e análise de dados, bem como a aplicação do protocolo de estudo de caso;

Capítulo 4 - Plataforma *ServiceNow* - Apresenta-se uma breve contextualização da empresa *ServiceNow*, fornecedora da plataforma de desenvolvimento *low-code/no-code*, que é empregada pela *organização*, objeto de estudo deste trabalho. Mais especificamente, explora-se o ambiente de desenvolvimento *App Engine Studio* (APS) e outros aspectos relevantes relacionados à plataforma;

Capítulo 5 - Impacto do *Low-code* na Organização - Apresenta as atividades realizadas no estudo de caso, descrevendo os resultados obtidos, as análises realizadas e as limitações encontradas;

Capítulo 6 - Conclusão e Trabalhos Futuros - Resume as principais conclusões deste trabalho, destaca as contribuições para a prática organizacional e apresenta recomendações para trabalhos futuros;

Apêndice A - Carta de Apresentação;

Apêndice B - Roteiro das Entrevistas;

Apêndice C - Carta de Agradecimento.

## 2 Referencial Teórico

### 2.1 Considerações Iniciais do Capítulo

Neste Capítulo, são apresentadas as bases conceituais deste trabalho, focando na abordagem *Low-code* e seu papel no apoio à Transformação Digital, utilizando o Design de Serviços como abordagem principal. Inicialmente, discute-se a Transformação Digital e como ela reconfigura operações empresariais para melhorar eficiência e adaptabilidade em um mercado dinâmico. São elucidados os fundamentos do Design de Serviços, destacando o *codesign* e a *cocriação*. Em seguida, a abordagem EUD e sua relevância na democratização do desenvolvimento de software e, a abordagem *low-code/no-code*, detalhando características, plataformas como *ServiceNow* e *Microsoft Power Apps*, benefícios, comparação entre *low-code* e *no-code*, e princípios fundamentais. Apresenta-se a aplicação do Design de Serviços no *low-code*, incluindo *cocriação* e *codesign*, enfatizando a colaboração contínua com os usuários finais. Discute-se o ciclo de vida do desenvolvimento *low-code* e suas áreas de atuação, como automação de processos e criação de aplicativos empresariais. Finalmente, apresenta-se um cenário do mercado *low-code/no-code*, suas tendências e os impactos futuros na indústria de desenvolvimento de software, destacando a transformação do cenário tecnológico atual.

### 2.2 Transformação Digital

A Transformação Digital (TD) é um processo abrangente e multifacetado que visa revisar e reinventar os processos centrais das organizações. Ela evolui de uma transição analógica para digital e envolve uma revisão completa de políticas, processos atuais e necessidades dos usuários. A TD não é apenas a digitalização de processos existentes, mas também a criação de novos serviços digitais, a reestruturação das interações com os clientes e a inovação nos modelos de negócios. Para [Verina e Titko \(2019\)](#), a TD não se limita à automação de processos já existentes,

mas se estende à reinvenção das formas de condução do negócio, interação com os clientes e a inovação dos modelos de serviços.

Segundo [Scupola e Mergel \(2022\)](#), a transformação digital inclui um esforço holístico que resulta em uma revisão completa dos serviços existentes e na criação de novos, com foco na satisfação das necessidades dos usuários, no desenvolvimento de novas formas de entrega de serviços e na expansão da base de usuários.

Organizações e governos têm investido em estratégias de Transformação digital. [Faro, Abedin e Cetindamar \(2021\)](#) observam que a TD no setor público ficou atrás do setor privado. As disrupções da era digital exigem um alto grau de flexibilidade nas operações e ajustes contínuos nas prioridades estratégicas, o que contrasta com a abordagem tradicional orientada para a estabilidade das organizações do setor público. Nas últimas duas décadas, governos em todo o mundo têm transitado do governo eletrônico rumo ao governo digital, empregando estratégias de transformação digital.

O relatório da [OECD \(2020\)](#) aponta os esforços do governo brasileiro para a transformação digital, com políticas públicas adequadas que promovam o crescimento econômico. Destacam ainda, a relevância da adaptação de um ambiente regulatório e de infraestrutura para facilitar essa transição. Observa-se que a alocação de 13% do orçamento de Pesquisa e Desenvolvimento para o Ministério de Ciência, Tecnologia, Inovação e Comunicação (MCTIC) é crítica para fomentar inovações tecnológicas que são essenciais para a transformação digital. Observa-se, ainda, o destaque da necessidade de uma gestão eficaz dos recursos e de uma coordenação estratégica entre vários atores governamentais e parceiros privados para maximizar o impacto das políticas públicas de digitalização e inovação tecnológica no Brasil. Além dos movimentos do atual Ministério da Gestão e Inovação de Serviços Digitais (MGI), com o Programa *Startup Gov.Br* e o emprego de especialistas (oriundos de concursos temporários) no apoio às estratégias de *startups* de governo ([VENSON; FIGUEIREDO; CANEDO, 2024](#); [MARSICANO et al., 2024](#)).

A Transformação Digital, ao propor a revisão e reinvenção de processos organizacionais, serve como base para a compreensão dos elementos que a suportam, como a digitização e a digitalização. Esses conceitos se destacam por impulsionar

a eficiência operacional e a criação de valor para os clientes, indo além da simples automação de processos.

A integração de serviços digitais com produtos tradicionais exemplifica como as organizações podem transformar suas operações e inovar nos modelos de negócio, alinhando-se às demandas de um mercado cada vez mais dinâmico e competitivo.

### 2.2.1 Digitização e Digitalização no contexto da TD

Os esforços para integrar serviços digitais com produtos tradicionais criam ofertas mais robustas e centradas no cliente. A digitização e a digitalização são elementos centrais nas estratégias de negócios e impulsionam a tomada de decisão baseada em dados e aumentam a eficiência operacional (KRYVINSKA; BICKEL, 2020).

No contexto da transformação digital é essencial distinguir claramente os conceitos de digitização e digitalização, pois cada um desempenha um papel fundamental na evolução do ambiente organizacional. Enquanto que a *digitização* se concentra na conversão de informações físicas para o formato digital, isto é, trata da conversão de informações analógicas em formato digital; a *digitalização* foca na transformação dos processos e modelos operacionais por meio de tecnologias digitais (CORESYSTEMS, 2024), isto é, está associada à utilização estratégica dessas informações para otimizar processos e modelos de negócios, com o uso da tecnologia (CORESYSTEMS, 2024). Ao unir essas dimensões, possibilita-se a criação de soluções inovadoras e personalizadas, promovendo maior eficiência operacional e novas formas de relacionamento entre empresas e clientes (KRYVINSKA; BICKEL, 2020).

Ao unir essas dimensões, possibilita-se a criação de soluções inovadoras e personalizadas, promovendo maior eficiência operacional e novas formas de relacionamento entre empresas e clientes (KRYVINSKA; BICKEL, 2020).

Os autores Minaya, Avella e Trespalacios (2024) apresentam um arcabouço teórico abrangente, demonstrando como a integração entre serviços e produtos digitais permite ganhos de competitividade na indústria. De maneira complementar,

Jia et al. (2024) destacam que a digitalização, ao ser aplicada de forma estruturada, possibilita a criação de plataformas digitais que ampliam a captura e análise de dados, transformando informações brutas em serviços personalizados para os clientes.

Para compreender melhor as diferenças e complementaridades entre esses conceitos, a Tabela 1 apresenta uma comparação entre digitização e digitalização. Essa diferenciação é essencial para que as organizações possam delinear estratégias eficientes de transformação digital, garantindo que cada etapa desse processo seja implementada de maneira adequada.

Tabela 1 – Comparação entre Digitização e Digitalização (CORESYSTEMS, 2024).

Característica	Digitização (Digitization)	Digitalização (Digitalization)
Definição	Conversão de informações físicas para o formato digital.	Uso de tecnologias digitais para transformar processos e modelos de negócios.
Escopo	Operacional e tecnológico.	Estratégico e organizacional.
Exemplo	Escanear documentos e armazená-los digitalmente.	Automatizar fluxos de trabalho para eliminar o uso de papel e integrar processos.
Impacto	Eficiência operacional sem mudar a estrutura organizacional.	Redesenho de processos, novos modelos de negócio e inovação.
Relacionamento	Primeiro passo para a digitalização.	Passo intermediário para a transformação digital.

A compreensão dessas distinções permite que as organizações desenvolvam estratégias eficazes de TD, utilizando a digitização como base inicial e a digitalização como meio de otimização, sendo ambas focadas na inovação e na competitividade (JIA et al., 2024).

A mudança estratégica promovida pela transformação digital envolve não só a automação, mas uma redefinição completa de como as empresas operam, se relacionam com os clientes e inovam nos modelos de negócios (VENDRELL-HERRERO et al., 2024). A digitalização não é apenas uma ferramenta, mas sim um pilar estratégico que permite decisões mais informadas e maior eficiência operacional.

Nesse contexto, a eficácia da TD é fundamental para governos e mercados que visam a inovação e a competitividade sustentável na era digital (VENDRELL-

[HERRERO et al., 2024](#); [KRYVINSKA; BICKEL, 2020](#)). Os estudos apresentados, mostram que os resultados reforçam a importância de investir no capital humano como fator estratégico para maximizar os benefícios da TD, especialmente em cenários que demandam inovação e adaptação contínuas. A adoção de tecnologias digitais redefine não só a entrega de produtos, mas transforma relações de mercado e cadeias de valor.

Essa capacidade de adaptação não só acelera a TD, mas também alinha a tecnologia com os processos de negócios e com as demandas do mercado. O desenvolvimento *low-code* ([SANCHIS et al., 2019](#); [BOCK; FRANK, 2021](#)), facilita essa integração ao reduzir a complexidade técnica necessária para desenvolver soluções e incorporar novas tecnologias no processo de desenvolvimento, permitindo que as organizações sejam mais ágeis e responsivas ao mercado. A eficácia desta TD, é fundamental para governos e mercados que visam a inovação e a competitividade sustentável na era digital ([VENDRELL-HERRERO et al., 2024](#); [KRYVINSKA; BICKEL, 2020](#)).

A TD não apenas otimiza processos e modelos de negócios, mas também redefine as interações entre organizações e usuários finais. No entanto, para que essa transição ocorra de forma fluida e eficiente, é essencial considerar a experiência do usuário e a adaptação dos serviços às suas necessidades. Nesse sentido, o Design de Serviços surge como um componente estratégico fundamental, permitindo que as organizações estruturem e aprimorem suas ofertas digitais de maneira intuitiva e centrada no cliente.

Ao integrar princípios do Design de Serviços nas organizações (privadas e públicas), pode-se garantir que a transição para modelos baseados em serviços seja não apenas tecnologicamente viável, mas também alinhada às expectativas e à jornada do usuário.

## 2.3 Design de Serviços

O Design de Serviços (DS) (*Service Design*) é uma abordagem inovadora para todo o contexto do mercado contemporâneo que visa facilitar proposições de valor apoiando os usuários a cocriar valor em seus devidos contextos ([TRIS-](#)



CHLER; TRISCHLER, 2022). O DS é uma abordagem que estrutura e otimiza a experiência dos usuários em serviços, considerando aspectos como usabilidade, jornada do cliente e interações entre atores do ecossistema (STICKDORN; LAWRENCE; HORMESS, 2022).

Stickdorn, Lawrence e Hormess (2022) complementam essa ideia destacando uma definição apontada como a mais popular do DS apresentada por Megan Erin Miller: *“O design de serviço ajuda as organizações a enxergarem seus serviços pela perspectiva do cliente. Ele promove experiências de serviço fluidas e de qualidade, ancorando-se no design thinking e oferecendo um processo criativo e centrado no ser humano para a melhoria de serviços e o projeto de novos serviços. Por meio de métodos colaborativos que envolvem clientes e equipes de serviço, o design de serviço auxilia as organizações a obterem uma compreensão verdadeira e completa de seus serviços, possibilitando melhorias holísticas e significativas”*.

Essa abordagem não foca apenas na transição de produtos para serviços, mas na *criação de serviços eficazes e desejáveis desde a concepção*. Stickdorn, Lawrence e Hormess (2022) destacam ainda que, o DS ajuda organizações a enxergarem seus serviços pela perspectiva do cliente, promovendo melhorias holísticas e significativas por meio de processos colaborativos. Demais autores ressaltam que o DS se distingue por reconhecer que a experiência do usuário é subjetiva e definida fenomenologicamente pelo usuário, baseando-se em atividades de cocriação de valor em contextos multi-atores (KAGAN, 2021), inclusive nas estratégias de governo, considerando as interações a partir do emprego do codesign e da cocriação (FIGUEIREDO et al., 2024a; FIGUEIREDO et al., 2024b).

Para criar interações significativas e personalizadas para os usuários, alguns pilares são fundamentais, como a empatia; a colaboração; e uma visão holística. A **Empatia** implica em compreender profundamente as necessidades e emoções dos usuários; a **colaboração**, que integra múltiplas perspectivas para enriquecer o processo de design; e uma **visão holística**, que assegura a coesão de todos os elementos do serviço, são cruciais para o desenvolvimento de soluções proativas que satisfaçam e até antecipem as expectativas dos usuários. Esses aspectos, ao serem enfatizados, permitem às organizações compreender melhor e aprimorar as experiências dos usuários, promovendo práticas de design de serviços verdadeira-

mente centradas no usuário e facilitando a criação de valor em contextos complexos e interativos (TRISCHLER; PERVAN; SCOTT, 2017; LAITINEN; KINDER; STENVALL, 2018).

Como abordagens centradas no ser humano para solução de problemas complexos e fomento da inovação, existem metodologias que compartilham filosofias que colocam a empatia pelo usuário e a colaboração interdisciplinar no centro de suas práticas, além de uma postura de experimentação e prototipagem rápida, o *codesign*, com a expansão da inclusão dos usuários como co-criadores no processo de design, emerge como uma estratégia poderosa, e a *cocriação*, que permite que as vozes dos usuários sejam diretamente integradas no desenvolvimento de soluções, garantindo que os resultados finais não apenas atendam às necessidades dos usuários, mas também sejam enriquecidos por suas perspectivas únicas e experiências vividas.

Dada a visão centrada no usuário, bem como a importância do processo da cocriação de valor dentro de um contexto complexo e interativo, Trischler e Trischler (2022) e Fox et al. (2022) complementam essa visão com o conceito da prática ágil de *codesign*. Enquanto Trischler e Trischler (2022) argumenta sobre a complexidade das interações durante um desenvolvimento de uma solução no serviço público, Fox et al. (2022) demonstra práticas de desenvolvimento colaborativo em um cenário de cuidados à saúde, usando plataformas *low-code* para adaptar rapidamente as tecnologias às necessidades dos usuários. Nessas interações, experiências são o resultado das atividades do usuário no micro-nível, ou seja, contexto do cotidiano dos usuário, e entendidas em uma esfera mais geral pelos desenvolvedores do serviço, durante as quais ele ou ela integra recursos de várias fontes, incluindo tecnologia digital, outras organizações e indivíduos dentro de um ecossistema de serviços.

Considerando o cenários de governos, Mergel, Ganapati e Whitford (2021) destacam que *os designers de serviços utilizam métodos etnográficos para compreender as necessidades dos usuários ao longo da jornada que fazem para acessar um serviço público. Eles entrevistam proprietários de processos (para entender os requisitos formais com base na lei) e usuários internos e externos (para compreender suas experiências ao longo de toda a jornada do usuário)*. Esse processo possibi-

lita revelar pontos problemáticos, mas também identificar aspectos que funcionam bem, criando oportunidades para projetar um melhor serviço público a partir da perspectiva do usuário. Além disso, enfatizam que *as etapas de concepção do serviço promovem inclusão e transparência – não apenas para os **cidadãos**, mas também para os **funcionários públicos**, que são membros centrais da equipe do projeto e participam ativamente dos processos de tomada de decisão*. Essas práticas, em adição à aplicação de plataformas *low-code*, oferecem agilidade na prototipagem e transparência no desenvolvimento, engajando tanto cidadãos quanto funcionários públicos nos processos de criação e melhoria de serviços.

Nesse cenário, plataformas *low-code* se apresentam como um viabilizador formidável, fornecendo um ambiente ágil para implementar as soluções projetadas pelo DS, permitindo que ideias sejam rapidamente transformadas em protótipos e sistemas funcionais. Plataformas *low-code*, facilitam a prototipagem e a adaptação iterativa às necessidades dos usuários. Por exemplo, em um ambiente *low-code*, o uso de ferramentas visuais e a automação de processos permitem que equipes multidisciplinares implementem rapidamente soluções cocriadas com usuários finais.

Dessa forma, o DS não apenas contribui para a criação de experiências centradas no ser humano, mas também encontra nas tecnologias *low-code* um aliado poderoso para implementar rapidamente as ideias e conceitos desenvolvidos, alinhando-se diretamente aos objetivos da transformação digital. Ademais, ao enfatizar a centralidade no usuário e a cocriação de soluções, estabelece uma base sólida para metodologias que buscam ampliar o envolvimento dos usuários finais no desenvolvimento de tecnologias. Nesse sentido, o *Desenvolvimento pelo Usuário Final* surge como uma extensão natural, permitindo que os próprios usuários, muitas vezes sem formação técnica, personalizem e criem soluções de software alinhadas às suas necessidades específicas.

## 2.4 Desenvolvimento pelo Usuário Final - EUD

O Desenvolvimento pelo Usuário Final (em inglês, *End User Development* (EUD)) é uma **área** emergente que possibilita aos usuários finais, muitas vezes sem formação em programação, participarem ativamente do desenvolvimento e perso-

nalização de software. O EUD tem como objetivo principal capacitar pessoas para empregar tecnologias avançadas de forma flexível, permitindo-lhes adaptar sistemas às suas necessidades, segundo [Lieberman, Paternò e Wulf \(2006\)](#). Isso engloba desde a configuração simples até a criação de novos aplicativos, utilizando métodos, técnicas e ferramentas que simplificam o desenvolvimento de software por não-profissionais ([MACÍAS, 2012](#)). A oferta de ferramentas de software aumenta a eficiência individual, mas também pode contribuir significativamente para a inovação organizacional.

O EUD capacita os usuários finais, que frequentemente não são profissionais da computação (programadores), a modificar e criar softwares para atender às suas necessidades específicas, utilizando abordagens intuitivas e visuais. Isso é particularmente relevante no ambiente de negócios moderno, em que a capacidade de adaptação rápida às mudanças de mercado e a personalização de ferramentas são imperativas. Segundo [Paternò \(2013\)](#), o desenvolvimento *low-code* pode ser entendido como uma prática EUD, já que ambos compartilham um objetivo comum: simplificar a tecnologia para habilitar uma maior agilidade e inovação no local de trabalho, refletindo a necessidade de ferramentas que democratizam o processo de desenvolvimento.

A transformação digital em organizações frequentemente envolve a implementação de tecnologias que agilizam processos e permitem a inovação contínua ([SANCHIS et al., 2019](#)). O EUD desempenha um papel crucial nesta transformação, pois *democratiza* a criação de software e permite que os usuários finais adaptem as ferramentas tecnológicas às suas necessidades imediatas, refletindo diretamente nos objetivos estratégicos da empresa. Essa capacidade de adaptação não só acelera a transformação digital, mas também alinha a tecnologia mais estreitamente com os processos de negócios e as demandas do mercado ([MELO; FIGUEIREDO; VENSON, 2018](#)). O desenvolvimento low-code ([BOCK; FRANK, 2021](#)), como uma expressão do EUD, facilita essa integração ao reduzir a complexidade técnica necessária para desenvolver soluções, permitindo que as organizações sejam mais ágeis e responsivas. Portanto, o EUD pode ser visto como um facilitador chave na transição para ambientes de negócios digitalmente avançados e adaptáveis, onde a inovação liderada pelo usuário final é uma vantagem competitiva significativa.

A crescente complexidade dos ambientes de negócios e a constante mudança nas necessidades dos usuários têm impulsionado o desenvolvimento de plataformas de EUD que oferecem interfaces mais acessíveis e intuitivas. Tais plataformas permitem que os usuários finais não apenas ajustem, mas também expandam suas funcionalidades sem a necessidade de conhecimentos avançados em programação (KO et al., 2011). Isso tem levado a uma nova forma de inovação dirigida pelo usuário, onde a solução de problemas e as melhorias não são mais exclusividade dos desenvolvedores de software.

O EUD apresenta uma série de desafios e oportunidades para a pesquisa em Interação Humano-Computador (IHC), Engenharia de Software e Trabalho Cooperativo Suportado por Computador (em inglês, *Computer Supported Cooperative Work*, CSCW). O envolvimento dos usuários finais no design e no desenvolvimento de software exige uma compreensão das suas práticas cotidianas e capacidades. A pesquisa em EUD busca desenvolver ferramentas que sejam ao mesmo tempo poderosas e acessíveis, equilibrando a flexibilidade na customização com a usabilidade e segurança necessárias, evidencia (BERGVALL-KÅREBORN; WIBERG, 2013).

Este movimento em direção ao EUD enfatiza a importância de se considerar os usuários finais não apenas como consumidores de tecnologia, mas como participantes ativos em sua criação e evolução (LIEBERMAN; PATERNÒ; WULF, 2006). Cada vez mais, o EUD se posiciona como uma peça importante para a inclusão digital, a democratização do desenvolvimento tecnológico e a transformação digital como um todo, promovendo uma maior autonomia e empoderamento dos usuários em diversos contextos, ressaltando a influência do EUD na capacitação dos usuários para adaptar e inovar dentro de suas plataformas de uso, segundo Paternò (2013).

Por fim, EUD emerge também como uma abordagem poderosa para mitigar o fenômeno do *Shadow IT*, permitindo que usuários finais desenvolvam soluções tecnológicas alinhadas às políticas organizacionais e às necessidades reais de seus contextos. Ao oferecer ferramentas acessíveis e intuitivas, o EUD reduz a necessidade de que colaboradores recorram a soluções não autorizadas, promovendo maior controle e transparência no desenvolvimento de software. No entanto, esse mesmo

empoderamento apresenta o desafio de equilibrar autonomia com governança, garantindo que as inovações criadas pelos usuários finais sejam seguras, compatíveis com as diretrizes da organização e estrategicamente alinhadas.

## 2.5 *Shadow IT*

A palavra *Shadow*, em inglês, quer dizer sombra em português, o que remete o cérebro a uma imagem escura ou a uma figura formada pela falta de luz projetada sobre um objeto, ou resumindo, ao desconhecido. Esse termo é utilizado nas ciências psicológicas, como áreas desconhecidas do cérebro, ou ainda nas finanças como riscos desconhecidos (GASPAR; MONTEIRO, 2018).

Já no contexto de Tecnologia da Informação, o termo *Shadow IT* pode ser definido, segundo Behrens e Györy (2009), como um fenômeno representado pela parte física (*hardware*), programa (*software*) ou qualquer outra solução usada pelos funcionários dentro do ecossistema organizacional e que não recebeu nenhuma aprovação formal do departamento de TI.

As soluções desenvolvidas, usadas para processos de negócios, não são desenvolvidas apenas pelo departamento de TI das empresas. Os departamentos de negócios e usuários implementam de forma autônoma, coloquialmente denominado de forma "caseira", soluções de TI que não são incorporadas no processo de governança de TI dessas empresas. Esse fenômeno cada vez mais frequente caracteriza a *Shadow IT*. Existem oportunidades e riscos no uso de *Shadow IT* que desafiam as organizações e exigem abordagens para gerenciar o fenômeno. Um ponto inicial para obter indicações mensuráveis para o gerenciamento é evidenciar o motivo pelo qual ele emerge (GASPAR; MONTEIRO, 2018).

O *Shadow IT*, de acordo com Zimmermann e Rentrop (2014), é criado após decisões fundamentadas nos custos de transação. As áreas de negócio, identificam esses custos proibitivos na relação de troca entre os departamentos de negócios e de TI. Esse desalinhamento serve como a principal explicação do uso de *Shadow IT* (ZIMMERMANN; RENTROP, 2014). Os princípios da teoria dos custos de transação podem ser aplicados para desenvolver estruturas de governança para o gerenciamento de *Shadow IT*. Isso fortalece o vínculo entre a governança de TI

e o seu alinhamento aos negócios, expandindo a compreensão da integração de negócios nos domínios de TI de uma organização (ZIMMERMANN; RENTROP, 2014).

Com o surgimento de novas tecnologias, que promovam o desenvolvimento prático de soluções de software, como por exemplo, a computação em nuvem, contribui para a atenuação do *Shadow IT* nas empresas.

Nesse sentido, para Strong e Volkoff (2004), o *Shadow IT* relaciona-se com questões de conformidade, perda de tempo, lógica de negócios inconsistente, aumento dos riscos de perda ou vazamento de dados, desperdício de investimento. Isso pode causar impacto em diversos aspectos para a empresa, como na segurança da informação, conformidades na qualidade de software e desalinhamento organizacional. Além disso, para os autores, o *Shadow IT*, desabilita sistemas legados e oficiais das empresas.

Portanto, o *Shadow IT*, ao expor a lacuna entre as necessidades dos usuários e as soluções formais de TI, pode ser transformado de um risco em um ativo estratégico por meio do *low-code*. Conforme destaca Begonha, Kopper e Thirakul (2022), plataformas *low-code* oferecem uma abordagem estruturada para que usuários sem expertise técnica desenvolvam soluções rapidamente, substituindo práticas não autorizadas por desenvolvimentos alinhados às políticas organizacionais. Essa abordagem não apenas mitiga os riscos do *Shadow IT*, mas também promove uma TD inclusiva e adaptável, ao permitir inovação ágil sem comprometer a governança e a segurança (BEGONHA; KOPPER; THIRAKUL, 2022).

## 2.6 Abordagem Low e No-Code

O desenvolvimento *Low e No-Code*, para Rokis e Kirikova (2023), é uma abordagem moderna que fornece a capacidade de criar software usando ferramentas de aplicação visuais de desenvolvimento de aplicativos, componentes predefinidos e automações, a minimizar, assim, a dependência da codificação manual. Além disso, permite que os indivíduos com experiência limitada em programação contribuam ativamente para o desenvolvimento do software juntamente com os profissionais da tecnologia da informação. Pode-se dizer que essa abordagem acelera o processo de

desenvolvimento, promove a colaboração acentuada entre as equipes de desenvolvimento e a área de negócios e aumenta a capacidade da organização de responder às demandas de negócios e do mercado (KAGAN, 2021).

O escopo do desenvolvimento *low-code/no-code* vai além do domínio de uma aplicação específica, incorporando diferentes domínios como aplicações web, mobile, internet das coisas, marketing, entre outros. Esta abordagem permite que programadores desenvolvam aplicativos focando no *design* e na estética, tendo a funcionalidade como um dos principais pilares, sem o esforço, por vezes impeditivo, de escrever códigos. Destaca-se, assim, a importância de uma interface gráfica amigável e de design orientado a modelos para o desenvolvimento de software, que reforça a relevância da programação focada no usuário final (PATERNÒ, 2013). Além disso, os softwares desenvolvidos perante essa abordagem incorporam os princípios da programação, ao permitir que profissionais com formações diversas e níveis variados de experiência em desenvolvimento de software para envolvam-se nessas atividades (ROKIS; KIRIKOVA, 2023).

Além de melhorar a acessibilidade ao desenvolvimento de software, as plataformas *low-code/no-code* também desempenham um papel importante na democratização da inovação digital. Ao empoderar usuários não técnicos com ferramentas que reduzem a complexidade do desenvolvimento de software, estas plataformas fomentam uma cultura de inovação em toda a organização. Isso é especialmente importante em um contexto onde a agilidade e a capacidade de resposta rápida às mudanças do mercado são cruciais (KAGAN, 2021). Estas plataformas funcionam como catalisadores da transformação digital, ajudando as organizações a se adaptarem mais rapidamente às exigências de um mercado em constante evolução (BOCK; FRANK, 2021).

A integração da transformação digital mostra como adaptações estratégicas podem ser implementadas mais eficientemente quando as ferramentas são intuitivas e acessíveis a todos os membros da organização, segundo Vendrell-Herrero et al. (2024). Isso, consequentemente, fortalece a colaboração entre diferentes departamentos, aumentando a capacidade de inovação e a implementação de soluções que realmente atendem às demandas do consumidor e do mercado (NAQVI et al., 2023).



Ainda nessa linha, para [Rokis e Kirikova \(2023\)](#), a abordagem *low-code/no-code* engloba processos que, de maneira geral, utilizam pouco código, "arrastar e soltar componentes", a existência de *templates* que passaram previamente pelo o processo de *design*, ou seja, um caminho mais amigável para pessoas que não são desenvolvedores de software profissionais.

Em síntese, é possível definir a abordagem *low-code/no-code* como uma metodologia que aprimora o desenvolvimento rápido, flexível e iterativo, isto é, um desenvolvimento de software que permite a tradução rápida de requisitos de negócios por meio de recursos visuais de programação nas interfaces gráficas, levando em consideração a abstração visual e codificação mínima, envolvendo, também, profissionais com formação em diversas áreas e com pouca experiência em desenvolvimento de software ([ROKIS; KIRIKOVA, 2023](#); [KAGAN, 2021](#)).

### 2.6.1 Plataformas de desenvolvimento *Low e No-Code*

A abordagem *low-code/no-code* está intimamente associada às plataformas de desenvolvimento *low-code/no-code* (do inglês *Low-code and no-code development platforms*, LCNC), consoante as ideias apresentadas por [Rokis e Kirikova \(2023\)](#). É possível afirmar que essa abordagem é baseada em ferramentas *low-code/no-code*, que são representadas nas plataformas de desenvolvimento, uma vez que elas possibilitam o desenvolvimento de aplicações, *deploy*, gerenciamento do ciclo de vida do produto e gerenciamento da plataforma em si.

As plataformas são usadas na nuvem - fornecidas pro meio do modelo Software como Serviço (SaaS), ou ainda no local em que o desenvolvimento é realizado usando componentes visuais, predefinidos e customizados. Em linhas gerais, as plataformas de desenvolvimento *low-code/no-code* é formada por várias camadas e podem ser distinguidas da seguinte maneira do ponto de vista arquitetural ([ROKIS; KIRIKOVA, 2023](#)):

- **Camada de aplicação:** Essa camada é composta por ambientes gráficos que incluem ferramentas e *widgets* para a definição da interface do usuário, além de incluir mecanismos para autenticação e autorização. É nessa camada que os usuários interagem com a plataforma, utilizando-a para modelar e

implementar os produtos de software e especificar seus comportamentos e especificidades;

- **Camada de integração de serviços:** A função dessa camada é promover a integração necessária para a utilização de diversos serviços, como as interfaces de programação da aplicação (APIs) e mecanismos de autenticação;
- **Camada de integração de dados:** Camada responsável por permitir a integração, operação e manipulação de dados que podem ser oriundos de diversas fontes;
- **Camada de *deploy*:** Camada relacionada à implantação da aplicação desenvolvida em uma nuvem dedicada ou em ambiente local. Em colaboração com a camada de integração de serviços, a containerização e orquestração da aplicação são realizadas nesta camada.

Consoante a [Kagan \(2021\)](#), as plataformas *low-code/no-code* promovem a base para o fluxo de trabalho básico, por meio da manipulação de dados, portais na web para usuários. Todos os serviços são desenvolvidos com o uso mínimo de código, com a funcionalidade de arrastar e soltar componentes, tornando, assim, as plataformas eficientes para a entrega de produtos. Elas são capazes de serem usadas no design de processos, ao promover meios para organizar o fluxo de planejamento e comunicação do time, com *workflows* e integrações especiais para envolver todos no desenvolvimento do produto.

Conforme [Rokis e Kirikova \(2023\)](#), as principais funcionalidades dessas plataformas são apresentadas na Tabela 2.

Tabela 2 – Algumas das *features* suportadas por plataformas LCNC, adaptado de (ROKIS; KIRIKOVA, 2023)

Características de Nível Superior	Características	Descrição
<b>Suporte à modelagem de requisitos</b>	<ul style="list-style-type: none"> <li>Ferramentas de gestão de requisitos</li> </ul>	<ul style="list-style-type: none"> <li>Incorpora funcionalidades para a especificação de requisitos.</li> <li>As plataformas geralmente fornecem ferramentas para o gerenciamento de requisitos, incluindo coleta de dados, seleção de critérios, uso de diagramas e integração desses requisitos nos planos de sprints.</li> <li>Essa característica é essencial para garantir a implementação correta dos requisitos, bem como para sua rastreabilidade e verificação.</li> </ul>
<b>Ferramentas de desenvolvimento visual</b>	<ul style="list-style-type: none"> <li>Designer visual</li> <li>Abordagem de arrastar e soltar</li> <li>Ambiente de desenvolvimento integrado</li> <li>Formulários</li> </ul>	<ul style="list-style-type: none"> <li>A partir do designer visual, todos os aspectos do software são desenvolvidos selecionando, organizando, configurando e conectando componentes.</li> <li>Uma abordagem de “arrastar e soltar” permite que os desenvolvedores criem aplicativos simplesmente agarrando e arrastando os elementos necessários para as posições desejadas na interface.</li> <li>Um ambiente de desenvolvimento integrado dedicado para os desenvolvedores por todo o ciclo de vida do desenvolvimento.</li> <li>Inclui dashboards, formulários de customização, pesquisas, checklists e mais, para melhorar a interface do usuário e a experiência do usuário.</li> </ul>
<b>Componentes de codificação avançados</b>	<ul style="list-style-type: none"> <li>Componentes avançados de codificação</li> </ul>	<ul style="list-style-type: none"> <li>Um recurso que permite alcançar uma personalização mais extensa para construir uma extensa interface de usuário e experiência de usuário, através de algoritmos personalizados suportando várias tecnologias.</li> </ul>
<b>Automação de Desenvolvimento</b>	<ul style="list-style-type: none"> <li>Geração automática</li> <li>Automação de detalhes de baixo nível</li> </ul>	<ul style="list-style-type: none"> <li>Uma característica que permite a geração automática de aplicações, por exemplo, gerando automaticamente uma aplicação a partir de uma planilha.</li> <li>Uma característica que automatiza preocupações de baixo nível no desenvolvimento de aplicações, como mapeamento objeto-relacional, balanceamento de carga, integridade de dados, otimização de consultas, serviços, mensagens, autenticação e mais.</li> </ul>
<b>Suporte ao Desenvolvimento Colaborativo</b>	<ul style="list-style-type: none"> <li>Colaboração online</li> <li>Colaboração offline</li> </ul>	<ul style="list-style-type: none"> <li>Mecanismos para trabalho simultâneo e colaboração por múltiplos desenvolvedores e gerenciamento de conflitos em tempo de execução.</li> <li>Mecanismos que suportam múltiplos desenvolvedores trabalhando localmente em modo offline e submetendo mudanças ao servidor em que mudanças são mescladas.</li> </ul>

O desenvolvimento de softwares e serviços com o uso de plataformas de desenvolvimento *low-code* surgem como um catalisador essencial nesse processo de transformação digital, permitindo que mesmo usuários não técnicos possam contribuir para a criação de soluções digitais. Essas plataformas simplificam o desenvolvimento de aplicativos, reduzindo a complexidade do código e acelerando a entrega de projetos (SANCHIS et al., 2019), (BOCK; FRANK, 2021). Isso não só democratiza a inovação tecnológica dentro das organizações, mas também potencializa a agilidade e adaptabilidade necessárias para responder rapidamente às mudanças do mercado e às necessidades dos clientes. A integração de plataformas *low-code* na indústria de manufatura, por exemplo, mostra como a transformação digital pode ser acelerada, permitindo que as empresas otimizem suas operações e introduzam novos serviços de forma mais eficiente e com menor custo.

### 2.6.2 Benefícios do *low-code/no-code*

As empresas precisam, de fato, ter respostas rápidas e contundentes frente às mudanças frequentes do mercado no hodierno. Com o intenso aumento da complexidade das operações internas, buscam-se novos meios de desenvolvimento de software em ambientes que forneçam ferramentas que visem à facilitar o processo de automação de tarefas nas companhias. As plataformas de desenvolvimento *low-code* cumprem essa função, uma vez que elas possibilitam aos desenvolvedores de software criar aplicações a partir de interfaces amigáveis e facilmente configuráveis, em detrimento a programação tradicional (SANCHIS et al., 2019).

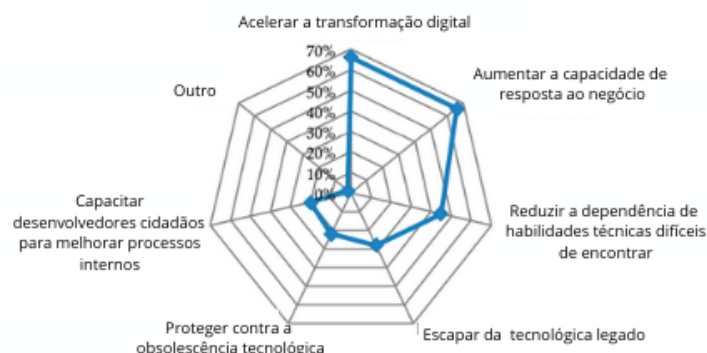
As plataformas de desenvolvimento *low-code/no-code* facilitam o desenvolvimento de protótipos do produto de software, uma vez que eles são relativamente rápidos de desenvolver, segundo Kagan (2021). Ademais, essas plataformas oferecem suporte à variados tipos de necessidades e de demandas oriundas de diferentes organizações e de negócios. Por meio de várias *features*, como, por exemplo, *websites*, *workflows*, relatórios interdepartamentais, modelagem de base de dados, entre outros. Kagan (2021) reforça que as plataformas de desenvolvimento *low-code* surgem como uma solução promissora devido à sua capacidade de desenvolver aplicações de forma rápida e eficiente, a economizar tempo e custos.

As plataformas de desenvolvimento *low-code* são capazes de desenvolver aplicativos operacionais completos, ou ainda, requisitar códigos adicionais para situações específicas. Além disso, elas abarcam todo o processo de modelagem da base de dados, fluxo de processos de negócios e interfaces de usuários, como, por exemplo, aplicações web. Este processo de desenvolvimento ágil e flexível é fundamental para adaptar-se rapidamente às exigências do mercado e às necessidades dos usuários (CABOT, 2020). Nesse cenário, elas são disponibilizadas em serviços na nuvem e têm, como principais aspectos, a velocidade com a qual os desenvolvedores e os clientes testam e obtêm *feedbacks*, para atuarem de maneira incremental e iterativa em cima do produto final (SANCHIS et al., 2019).

Além disso as plataformas *low-code* relacionam-se diretamente com o conceito de *Shadow IT*, que, de acordo com a teoria de (ZIMMERMANN; RENTROP, 2014), surge em decorrência do alto custo de transação associado aos processos formais de desenvolvimento e aquisição de soluções de TI, o que frequentemente leva setores não especializados a buscar alternativas independentes para atender às suas necessidades, o que permite que usuários finais desenvolvam aplicações com menor dependência de equipes especializadas e processos burocráticos, gerando desalinhamento na política de governança de tecnologia da empresa. Como destacado por Begonha, Kopper e Thirakul (2022), o *low-code* transforma o *Shadow IT* de uma ameaça à governança de TI em um ativo estratégico, capacitando colaboradores a criar soluções alinhadas às políticas corporativas. Essa abordagem não apenas mitiga os riscos associados ao *Shadow IT*, mas também promove maior agilidade e eficiência, ao permitir que as empresas entreguem valor mais rapidamente, com custos menores e maior conformidade.

Na Figura 1, apresentam-se os resultados obtidos por meio de uma pesquisa respondida por mais de 3300 profissionais de TI de diferentes continentes quanto aos principais motivos para o uso de plataformas de desenvolvimento *low-code*. Dos respondentes, 66% escolheram acelerar a transformação digital e aumentar a capacidade de resposta ao negócio como os principais motivos para o uso ou intenção de uso de plataformas de desenvolvimento *low-code*; 45% dos profissionais pesquisados apontaram para a redução da dependência em habilidades técnicas difíceis de contratar.

Figura 1 – Principais razões para se usar plataformas *low-code* (Adaptado)(SANCHIS et al., 2019).



Em resumo, as plataformas de desenvolvimento *low-code/no-code* emergem como ferramentas essenciais no cenário da Indústria digital contemporânea, ao oferecer uma resposta ágil e adaptável às exigências de um mercado em constante transformação (KAGAN, 2021). Essas plataformas não apenas simplificam e aceleram o processo de desenvolvimento de software, mas também democratizam a criação tecnológica ao reduzir a necessidade de habilidades técnicas avançadas (BOCK; FRANK, 2021). Ao permitir que os desenvolvedores e as organizações prototipem e iterem rapidamente sobre suas soluções, elas facilitam um melhor alinhamento com as necessidades do usuário final e, muitas vezes, incluindo-o nesse processo e fomentando o *codesign*, gerando uma maior eficiência operacional (SANCHIS et al., 2019). A capacidade de acelerar a transformação digital e responder de forma efetiva às demandas do negócio, posiciona a abordagem *low-code/no-code* como uma vantagem competitiva significativa, evidenciando sua relevância crescente para sustentar a inovação e a competitividade no ambiente de negócios moderno.

### 2.6.3 Comparação entre *low-code* e *no-code*

Em relação ao uso de plataformas de desenvolvimento *low-code* e *no-code*, com interfaces gráficas de usuário ou interfaces visuais para habilitar e produzir aplicativos de negócios, a abordagem *low-code* ainda permite incluir codificação em alguns casos, enquanto o *no-code* não exige nenhuma codificação, de acordo com

exposto por [Veerasamy \(2022\)](#). A abordagem *no-code* destina-se exclusivamente a desenvolvedores não especializados, enquanto a abordagem *low-code* destina-se tanto a desenvolvedores não especializados quanto a desenvolvedores especialistas em tecnologia da informação.

As plataformas *low-code* e *no-code*, ainda podem exigir o desenvolvimento de aplicações maiores e mais complexas. Elas podem combinar abordagens *low-code* e *no-code* e são utilizadas por empresas com visão de futuro para melhorar a flexibilidade e o controle sobre o ciclo de desenvolvimento. Três fatores principais distinguem as plataformas de desenvolvimento *no-code* das plataformas de desenvolvimento *low-code*, conforme mostrado nos seguintes tópicos, ilustrados em [Veerasamy \(2022\)](#):

- **Habilidades de codificação:** As plataformas *no-code* estão disponíveis para todos os usuários finais, tendo em vista que nenhuma habilidade técnica é necessária para o usuário. Por outro lado, desenvolvedores com experiência em codificação e que possam operar em uma plataforma para facilitar e agilizar o processo de desenvolvimento são necessários para o desenvolvimento em plataformas *low-code*;
- **Design centrado em quem desenvolve:** Nas plataformas *no-code*, o usuário final escolhe o design de um aplicativo usando a lógica de arrastar e soltar. Já em relação ao suporte oferecido nas plataformas *low-code* é usado um paradigma de desenvolvimento semelhante, mas depende mais da codificação para lidar com a arquitetura central de um aplicativo e configurações mais específicas;
- **Interface do usuário:** Para facilitar o design do aplicativo, a abordagem *no-code* depende de uma camada de interface do usuário pré-definida. As plataformas *low-code* fornecem mais flexibilidade nesse sentido, ao passo que precisam de maiores necessidades de codificação. Esse desenvolvimento inclui a capacidade de fazer codificação para fins de personalização ou complexidade do aplicativo, enquanto o desenvolvimento com a abordagem *no-code* não contém nenhuma codificação para certificar que os aplicativos sejam tão fáceis de usar quanto possível. Da mesma forma, as plataformas *low-code*

têm a opção de abrir, visualizar e alterar o código-fonte dos aplicativos, enquanto as plataformas *no-code* não. Minimizar o tempo de desenvolvimento do código é considerado uma vantagem pela maioria das plataformas de desenvolvimento *low-code/no-code*, e, além disso, a quantidade mínima de código-fonte reduz o tempo de desenvolvimento e torna a aplicação mais fácil de ser desenvolvida e entregue

Concluindo, a distinção entre as plataformas com abordagens *low-code* e *no-code* desempenha um papel fundamental na democratização do desenvolvimento de aplicações, tornando a tecnologia acessível tanto para usuários técnicos quanto para não técnicos. Enquanto as plataformas *no-code* oferecem uma solução totalmente visual e intuitiva para usuários sem habilidades técnicas, permitindo que desenvolvam aplicações sem escrever uma única linha de código, as plataformas *low-code* oferecem maior flexibilidade e controle, adequadas para desenvolvedores com diversos níveis de experiência técnica. Essas plataformas não apenas facilitam o desenvolvimento rápido de aplicações, como também permitem que organizações ágeis e com visão de futuro ajustem seus processos e fluxos de trabalho de forma eficaz, como bem exposto por [Veerasley \(2022\)](#). Assim, a escolha entre *low-code* e *no-code* deve ser guiada pela necessidade de customização, complexidade do projeto e o nível de expertise dos usuários envolvidos no desenvolvimento.

#### 2.6.4 Princípios do desenvolvimento *Low Code*

Para explorar os princípios do desenvolvimento *low-code* de uma revisão abrangente da literatura, é essencial entender os princípios fundamentais que regem essa abordagem inovadora de desenvolvimento ([ROKIS; KIRIKOVA, 2023](#)). As plataformas de desenvolvimento de *low-code* simplificam o processo de desenvolvimento de software ao reduzir a necessidade de codificação extensiva, permitindo, assim, o rápido desenvolvimento e implantação de aplicativos. Esses princípios servem como alicerce para utilizar de maneira eficaz as tecnologias de *low-code* nas organizações, promovendo maior agilidade nos negócios e inovação.

**1. Princípio do Desenvolvimento Visual e Abstração:** As plataformas de *low-code* utilizam intensamente ambientes de desenvolvimento visual, per-



mitindo aos usuários construir aplicações por meio de interfaces gráficas ao invés de métodos de codificação tradicionais. Essa abordagem não apenas simplifica o processo de desenvolvimento, mas também democratiza o desenvolvimento de aplicativos, tornando-o acessível a uma gama mais ampla de usuários, incluindo aqueles com habilidades mínimas de codificação. Esse princípio enfatiza a importância de abstrair complexidade através de modelos visuais, que se alinham com as capacidades da plataforma para facilitar funcionalidades de arrastar e soltar e templates pré-construídos (ROKIS; KIRIKOVA, 2023; CABOT, 2020).

**2. Princípio da Automação e Integração:** A automação é um componente crucial das plataformas de *low-code*, simplificando várias etapas do ciclo de vida do desenvolvimento, desde o design inicial até a implantação final. Essas plataformas integram ferramentas avançadas de automação que gerenciam processos de fundo, como integração de bancos de dados e conexões API, sem necessitar de intervenção manual (ROKIS; KIRIKOVA, 2023). Esse princípio garante que os desenvolvedores possam focar mais em projetar soluções que atendam às necessidades empresariais em vez de lidar com complexidades técnicas subjacentes (NAQVI et al., 2023).

**3. Princípio da Colaboração e Acessibilidade:** O desenvolvimento de *low-code* promove naturalmente a colaboração entre profissionais de TI e não-TI ao fornecer ferramentas fáceis de usar e entender. Esse ambiente colaborativo aprimora o alinhamento entre os requisitos empresariais e a implementação técnica, levando ao desenvolvimento de soluções mais eficazes e eficientes. A acessibilidade das plataformas de *low-code* permite que uma gama diversificada de usuários contribua para o processo de desenvolvimento, fomentando inovação e prototipagem rápida em vários departamentos (VENDRELL-HERRERO et al., 2024).

**4. Princípio da Melhoria Contínua e Adaptabilidade:** As plataformas de *low-code* são projetadas para serem flexíveis e adaptáveis, permitindo que as organizações respondam rapidamente a mudanças no mercado e necessidades empresariais em evolução. A melhoria contínua é facilitada pelo desenvolvimento iterativo e atualizações frequentes, que são essenciais para manter a relevância e a eficiência das aplicações empresariais. Este princípio destaca a necessidade de plataformas que suportem modificações e extensões fáceis em aplicações existentes,

permitindo que as empresas inovem e se adaptem sem começar do zero (SANCHIS et al., 2019).

Em conclusão, os princípios do desenvolvimento de *low-code* giram em torno da simplificação do processo de desenvolvimento de software, aprimorando a colaboração em vários níveis organizacionais e fornecendo a agilidade necessária para se adaptar a ambientes empresariais em mudança. Esses princípios não apenas facilitam o desenvolvimento rápido de aplicações, mas também garantem que as soluções sejam robustas, escaláveis e alinhadas com os objetivos estratégicos da organização. Essa abordagem reflete uma mudança significativa na forma como o software é desenvolvido e implantado, enfatizando a velocidade, eficiência e envolvimento do usuário no processo de desenvolvimento.

### 2.6.5 DS no desenvolvimento *low-code*

Um aspecto particularmente importante na Indústria 4.0 é a importância do DS na digitalização, reforça Trischler e Trischler (2022), em que a interação entre tecnologias avançadas e os serviços se torna cada vez mais complexa e integrada. A implementação eficaz do DS requer uma compreensão profunda das necessidades dos usuários finais, a capacidade de prototipar e testar rapidamente novas ideias, e a agilidade para iterar com base no feedback do usuário. Adicionalmente, as reflexões em Cabot (2020) e Sanchis et al. (2019) ilustram como plataformas *low-code* facilitam essa agilidade e adaptabilidade, permitindo uma transformação digital mais rápida e inclusiva, essencial para manter a competitividade na era moderna.

Para observar esse processo, que envolve um estudo longitudinal multi-caso, Iriarte et al. (2023) observou a aplicação prática do DS em empresas reais. Ao longo de dois anos e meio, a partir de uma metodologia promovida por Malmberg (2017), as práticas de DS de 18 companhias de tecnologia e manufatura, tais como a *ConstrucSys* e a *GearbSys*, foram estudadas utilizando o método de pesquisa sugerido por meio do DS. Esta abordagem permitiu aos pesquisadores não apenas observar as práticas de design de serviço, mas também participar ativamente do desenvolvimento e da implementação das proposições de valor de serviços avançados. Os resultados revelaram que a adoção do DS facilitou uma melhoria média de 30% na

eficiência de integração de novas tecnologias digitais e uma melhoria de 25% na satisfação do cliente com os serviços.

Essa pesquisa também indicou que 85% das empresas participantes reportaram um aumento significativo na colaboração interna, e 75% observaram uma melhoria na adaptação às mudanças tecnológicas e nas demandas do mercado, ilustrando assim o impacto prático e quantificável do design de serviço na transformação digital das empresas envolvidas. A Figura 2, ilustra as diferentes fases de um projeto de pesquisa ou de um processo de *Design* de serviço na digitalização de serviços.

Figura 2 – Processo de *Design de Serviço* na digitalização de serviços *low-code* (Adaptado)([IRIARTE et al., 2023](#)).



Nesse ínterim, o impacto do *Design* de serviço na digitalização de serviços em empresas contemporâneas como um todo fica evidente. Por meio do DS, com a incorporação de ferramentas como mapas da jornada do cliente e *blueprints* de serviço, as empresas são capazes de adotar uma abordagem holística que não apenas melhora a colaboração e comunicação interna e externa, mas também otimiza a oferta e entrega de serviços avançados. Isso permite também uma revisão crítica e estratégica das proposições de valor existentes e incentiva a inovação focada no cliente. A integração das plataformas *low-code* neste contexto acelera a implementação de soluções digitais, permitindo que as organizações respondam mais rapidamente às necessidades emergentes e iterem sobre os serviços com maior efici-

ência (CABOT, 2020), (SANCHIS et al., 2019). Assim, a sinergia entre o *Design* de serviço e o desenvolvimento *low-code* torna-se fundamental para a efetiva transformação digital, facilitando uma cultura de inovação ágil que adapta as tecnologias às necessidades humanas em constante evolução.

### 2.6.6 Ciclo de vida

O desenvolvimento e entrega de software por meio de uma abordagem *low-code/no-code* compreende várias etapas de desenvolvimento de software a serem realizadas. Qualquer metodologia de desenvolvimento de software pode ser aplicada no desenvolvimento *low-code/no-code*, mas as organizações frequentemente usam práticas de entrega rápida. Os estágios típicos de desenvolvimento de softwares com a abordagem *low-code/no-code* são descritos com base em sete fases para que seja possível desenvolver aplicativos usando plataformas de desenvolvimento de *low-code*:

1. **Análise de requisitos e viabilidade:** Identificação e avaliação dos requisitos do sistema, incluindo análise de viabilidade técnica e econômica. Esta fase envolve a comunicação com os *stakeholders* para entender as necessidades do projeto e estabelecer objetivos claros e alcançáveis.
2. **Modelagem de dados:** O esquema de dados da aplicação é definido e configurado. Isso envolve estabelecer a estrutura das entidades, definir as suas relações, implementar restrições e identificar dependências dentro do modelo de dados;
3. **Definição da interface do usuário:** Nesta fase, os formulários, páginas e visualizações da aplicação são criados e as funções de usuário e mecanismos de segurança são configurados;
4. **Implementação de lógica de negócios e fluxos de trabalho:** Fluxos de trabalho de aplicativos e lógica de negócios são definidos e configurados;
5. **Integração de serviços externos:** As integrações necessárias com serviços de terceiros são estabelecidas;

6. **Teste e implementação:** Testes são realizados para garantir que a aplicação atende aos requisitos definidos e funciona corretamente em diversos cenários. Esta fase inclui testes unitários, de integração, de sistema e de aceitação. Após a validação, a aplicação é implementada no ambiente de produção.
7. **Feedback do cliente e recursos adicionais:** Após a implantação do aplicativo, o *feedback* das partes interessadas são recebidos e, se necessário, alterações e recursos adicionais são definidos para o ciclo subsequente;

O uso de conceitos ágeis no desenvolvimento com a abordagem *low-code/no-code* são, de fatos utilizados (SANCHIS et al., 2019). De acordo com Rokis e Kirikova (2023), a metodologia ágil e o desenvolvimento com a abordagem *low-code/no-code* estão em conformidade, uma vez que promovem uma metodologia de entrega iterativa e frequente, com envolvimento contínuo das partes interessadas. O uso de vários *frameworks* é descrito, a exemplo dos conceitos de *Extreme Programming* que são aplicados. Além disso, é recomendado o uso da *Lean-Inception*, juntamente com o *Scrum* e o *Kanban* para que a relação entre os estágios de desenvolvimento de softwares com a abordagem *low-code/no-code* e metodologias ágeis possam ser descritos seguindo essa sequência (ROKIS; KIRIKOVA, 2023):

1. **Análise de Requisitos:** Os requisitos relacionados ao software a ser desenvolvido são identificados, e ferramentas de gerenciamento de requisitos de plataforma podem ser utilizadas;
2. **Planejamento:** Realiza-se a análise e o planejamento da viabilidade, cronograma, interdependências, riscos e complexidades;
3. **Design da Aplicação:** Com base nos requisitos definidos, o design é especificado e revisado pelos *stakeholders* considerando arquitetura, modularidade e extensibilidade;
4. **Desenvolvimento da Aplicação:** A aplicação é desenvolvida utilizando recursos da plataforma de *low-code/no-code*, especificando interface do usuário, lógica de negócio, integração e outros elementos;

5. **Testes:** Testes são realizados para verificar a implementação dos requisitos dentro do software;
6. **Implantação:** A aplicação é implantada localmente ou na nuvem;
7. **Manutenção:** Manutenção é fornecida, uma vez que a aplicação é lançada, e, se a funcionalidade deve ser estendida, recursos adicionais podem ser adicionados, ou uma nova iteração pode ser iniciada;

A abordagem *low-code/no-code* oferece uma alternativa poderosa e ágil para o desenvolvimento de software, permitindo uma entrega mais rápida e iterativa. A integração dessa abordagem com as metodologias ágeis facilita a adaptação rápida às mudanças de requisitos e melhora o envolvimento dos *stakeholders* ao longo de todo o ciclo de desenvolvimento (SANCHIS et al., 2019; ROKIS; KIRIKOVA, 2023). As práticas recomendadas, tais como a aplicação dos princípios de *Extreme Programming*, *Lean-Inception*, *Scrum* e *Kanban*, são vitais para maximizar a eficiência e eficácia do desenvolvimento *low-code/no-code*, reiterando sua compatibilidade com as necessidades dinâmicas de projetos modernos de software. Esta abordagem não apenas simplifica o processo de desenvolvimento mas também empodera os desenvolvedores a entregar soluções robustas e personalizadas com rapidez e precisão.

### 2.6.7 Áreas de atuação

As áreas de atuação para um desenvolvimento com a abordagem *low-code/no-code* são identificadas em aplicações web, dispositivos móveis, serviços empresariais, processos de negócios e IoT (internet das coisas), segundo Rokis e Kirikova (2023). Outras áreas incluem saúde, educação, bancos de dados, tratamento de solicitações, sistemas de recomendação, manufatura, treinamento industrial, mídia social, processo, marketing, blockchain e inteligência artificial (SANCHIS et al., 2019). Isso pode ser visto na Figura 3, em que o gráfico de barras mostra a frequência com que cada domínio é abordado nos estudos primários revisados pelos pares, reforçando a ideia de que o desenvolvimento *low-code/no-code* não é restrito a um nicho, mas é utilizado em uma vasta gama de setores. Este dado corrobora a

argumentação de que as plataformas *low-code/no-code* são flexíveis e capazes de se adaptar às necessidades variadas de diferentes campos e indústrias (BUCAIONI; CICHETTI; CICOZZI, 2022).

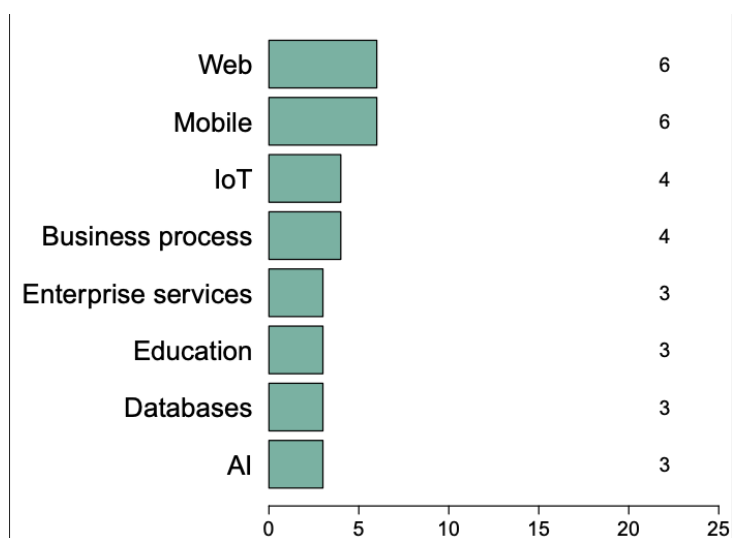


Figura 3 – Domínios dos estudos no que tange a abordagem *low-code* (BUCAIONI; CICHETTI; CICOZZI, 2022).

Um exemplo pode ser visto na pesquisa desenvolvida por Fox et al. (2022), que explorou o uso de um aplicativo para pessoas com demência, ilustra de maneira prática como os princípios de desenvolvimento *low-code* e o design ágil e iterativo podem ser aplicados para envolver diretamente os usuários finais no processo de design. Esse estudo exemplifica como o feedback contínuo dos usuários e de seus cuidadores pode ser essencial para adaptar a tecnologia às suas necessidades reais, resultando em um produto final que é não apenas funcional, mas também intuitivo e acessível para seu público-alvo.

Adicionalmente, as plataformas de desenvolvimento *low-code/no-code* têm sido reconhecidas por sua capacidade de reduzir drasticamente os ciclos de desenvolvimento de software. Essas plataformas permitem que mesmo usuários sem conhecimentos profundos em programação possam construir soluções complexas, graças à utilização de interfaces gráficas intuitivas e componentes pré-configurados (BOCK; FRANK, 2021; SANCHIS et al., 2019). Isso não apenas acelera o desen-

volvimento, mas também democratiza a tecnologia, permitindo que uma maior diversidade de profissionais contribua para a inovação tecnológica nas empresas.

A flexibilidade das plataformas *low-code/no-code* também se estende ao suporte para integrações complexas com outras tecnologias, como a Internet das Coisas (IoT) e sistemas de inteligência artificial. A capacidade de integrar facilmente essas tecnologias abre novas possibilidades para a automação de processos e a criação de sistemas cada vez mais inteligentes e conectados (PATERNÒ, 2013; KAGAN, 2021). Este aspecto é crucial para empresas que buscam inovação contínua e uma vantagem competitiva no mercado altamente dinâmico de hoje.

Em suma, a abordagem *low-code/no-code* tem se mostrado versáteis, abrangendo uma ampla gama de áreas de atuação que se estendem por diversos setores da indústria e serviços. Esta flexibilidade proporciona uma ponte natural para o *Design de Serviços* no desenvolvimento *low-code*, onde a ênfase na cocriação de valor, colaboração e empatia alinha-se perfeitamente com os objetivos de maximizar a eficiência e melhorar a experiência do usuário. Assim, o *low-code* não apenas facilita a implementação de soluções tecnológicas, mas também potencializa a inovação no design de serviços, permitindo que as organizações respondam de maneira ágil e eficaz às necessidades em constante evolução de seus clientes e mercados (KAGAN, 2021).

### 2.6.8 Mercado *low-code/no-code*

A necessidade de um rápido desenvolvimento e a constante iteração para com os produtos de software, de acordo com as ideias apresentadas por Bratincevic e Koplowitz (2021), tornou o uso das plataformas *low-code/no-code* uma abordagem fundamental no processo de transformação digital de diversas empresas. Os fornecedores desses ambientes têm procurado diferentes maneiras de aprimorar suas capacidades de atender segmentos mais especializados de casos de uso e diferentes personas de desenvolvedores.

O rápido desenvolvimento de aplicações e a constante iteração com softwares tornaram-se os pilares para o uso das plataformas *low-code/no-code*. Assim,



elas são uma abordagem de desenvolvimento de primeira classe (BRATINCEVIC; KOPLOWITZ, 2021).

Ao mesmo tempo, fornecedores de segmentos mais especializados estão amadurecendo suas capacidades em uma gama mais ampla de casos de uso e de personas de desenvolvedores – levando a uma convergência na automação de processos digitais e de mercados para o *low-code* (MATVITSKY; DAVIS; JAIN, 2024).

De acordo com Bratincevic e Koplowitz (2021) e de modo convergente com a teoria de Matvitsky, Davis e Jain (2024), o desafio oriundo da necessidade de padronização das plataformas é resumido em três pontos principais:

1. **Abordar padrões de desenvolvimento centrados em dados e em processos:** As necessidades do mercado variam desde a substituição básica de planilhas até a modernização central de fluxos de trabalho, orquestração de processos de longa duração (sempre com a experiência do usuário sólida e conectada em vários canais e em um cenário tecnológico heterogêneo). Em um cenário geral, nas plataformas *low-code/no-code*, os projetos simples devem ser fáceis e os projetos complexos devem ser eficientes;
2. **Fornecer ferramentas adequadas para diferentes perfis de desenvolvedores:** É necessário fornecer ambientes de desenvolvimento com ferramentas que atendam as habilidades de desenvolvedores profissionais em tecnologia da informação, assim como dos desenvolvedores cidadãos (conceito que é definido na Subseção 4.3.1), atendendo um espectro maior de colaboradores que irão atuar juntos na plataforma para o desenvolvimento de produtos de software;
3. **Atender aos requisitos específicos de infraestrutura, arquitetura e processo de desenvolvimento:** É importante atender aos aspectos específicos de infraestrutura de tecnologia da informação, como, exemplo, alguns fornecedores implantam as plataformas virtualmente e em qualquer lugar, enquanto outros oferecem apenas um modelo tradicional de software como serviço (SaaS). Há também os que fornecem todas as ferramentas necessá-

rias para os processos de desenvolvimento com foco na integração com as ferramentas *DevOps* existentes dos clientes. Dessa maneira, vale ressaltar a importância do aspecto de infraestrutura como código (IaaS) no uso das plataformas *low-code/no-code*.

Bratincevic e Koplowitz (2021) e Matvitsky, Davis e Jain (2024) apontam como líderes de mercado no fornecimento de plataformas de desenvolvimento *low-code/no-code*: *Microsoft Power Apps*, *ServiceNow*, *Mendix*, *Appian*, *OutSystems* e a *PegaSystems*.

Os tópicos a seguir mostram como essas empresas líderes de mercado relacionam-se com os principais pilares elencados anteriormente sobre a demanda da padronização da abordagem *low-code/no-code* no mercado:

### 1. *Microsoft Power Apps*

- a) **Abordar padrões de desenvolvimento centrados em dados e em processos:** Oferece uma plataforma abrangente com capacidades de integração destacadas, suportando uma vasta gama de necessidades de automação de processos e fluxos de trabalho de longa duração;
- b) **Fornecer ferramentas adequadas para diferentes perfis de desenvolvedores:** A plataforma suporta tanto desenvolvedores profissionais quanto desenvolvedores cidadãos, com ferramentas como a linguagem de desenvolvimento estilo *Excel* (Fx) e recursos de inteligência artificial projetados para desenvolvedores cidadãos (4.3.1);
- c) **Atender aos requisitos específicos de infraestrutura, arquitetura e processo de desenvolvimento:** *Power Apps* está disponível como um modelo tradicional de software como serviço (SaaS), integrado com serviços adjacentes do *Power Platform* e serviços úteis do *Azure*. No entanto, a necessidade de licenciamento separado pode causar confusão para os usuários;

Nessa linha, *Microsoft Power Apps* é parte do *Microsoft Power Platform*, oferecendo uma abordagem *low-code* para o desenvolvimento de aplicações

empresariais. A plataforma integra-se perfeitamente com o *Azure* e o *Microsoft 365*, permitindo que desenvolvedores criem soluções escaláveis com conectores prontos para diversas aplicações empresariais (MATVITSKYY; DAVIS; JAIN, 2024). A *Power Apps* suporta *Canvas Apps* e *Model-Driven Apps*, permitindo tanto um desenvolvimento altamente customizável quanto uma abordagem baseada em modelos de dados estruturados (JAIN; DAVIS; MATVITSKYY, 2024). No entanto, algumas funcionalidades avançadas e ferramentas de automação exigem licenciamento separado, o que pode gerar confusão e aumentar os custos para empresas que buscam uma solução completa (MATVITSKYY; DAVIS; JAIN, 2024).

## 2. *ServiceNow*

- a) **Abordar padrões de desenvolvimento centrados em dados e em processos:** Conhecida por suas soluções de gerenciamento de serviços de tecnologia, a *ServiceNow* oferece fortes capacidades de automação de processos e ferramentas simples para a extensão de objetos de dados canônicos, como exemplo o *App Engine Studio*;
- b) **Fornecer ferramentas adequadas para diferentes perfis de desenvolvedores:** A plataforma permite que desenvolvedores não técnicos façam e propaguem mudanças facilmente, apoiando a colaboração entre desenvolvedores profissionais e cidadãos (4.3.1) no principal ambiente de desenvolvimento, o *App Engine Studio*;
- c) **Atender aos requisitos específicos de infraestrutura, arquitetura e processo de desenvolvimento:** Oferecida através de um modelo tradicional de software como serviço (SaaS), a *ServiceNow* é forte para clientes existentes que desejam construir sobre a credibilidade de tecnologia da informação e a filosofia de fluxos de trabalho digitais da plataforma;

Isso posto, a *ServiceNow* permite o desenvolvimento de aplicações *low-code* e a automação de processos empresariais, aproveitando sua experiência consolidada em gestão de serviços de TI (ITSM) (MATVITSKYY; DAVIS; JAIN, 2024). Seu diferencial está na capacidade de automatizar fluxos de trabalho

complexos, integrando ferramentas de inteligência artificial e mineração de processos. A plataforma inclui o *App Engine Studio*, que simplifica a modelagem e implementação de soluções (JAIN; DAVIS; MATVITSKY, 2024). Além disso, a *ServiceNow* promove a colaboração entre desenvolvedores cidadãos e técnicos, facilitando a personalização de aplicações sem necessidade de conhecimento profundo em programação. Contudo, a dependência da infraestrutura *SaaS* da *ServiceNow* pode ser uma limitação para empresas que buscam maior controle sobre seus ambientes (MATVITSKY; DAVIS; JAIN, 2024)

### 3. *Mendix*

- a) **Abordar padrões de desenvolvimento centrados em dados e em processos:** *Mendix* oferece ferramentas excepcionais para modelagem e gestão de dados, com um catálogo de dados superior em termos de recursos e experiência do usuário;
- b) **Fornecer ferramentas adequadas para diferentes perfis de desenvolvedores:** A plataforma suporta uma ampla gama de personas de desenvolvedores, promovendo a colaboração entre a área de negócio e a de tecnologia. Suas credenciais *cloud-native* são superiores, oferecendo opções de implantação robustas;
- c) **Atender aos requisitos específicos de infraestrutura, arquitetura e processo de desenvolvimento:** Como uma plataforma *cloud-native*, *Mendix* é ideal para empresas que buscam uma solução para diversas necessidades de desenvolvimento *low-code*. No entanto, sua rede de parceiros de implementação é modesta em comparação com os concorrentes;

Dessa forma, a *Mendix* se diferencia pelo forte suporte ao desenvolvimento ágil, permitindo colaboração entre equipes de negócio e TI através de um ambiente de desenvolvimento visual avançado (JAIN; DAVIS; MATVITSKY, 2024). Com os ambientes *Mendix Studio* e *Mendix Studio Pro*, a plataforma oferece opções tanto para desenvolvimento visual quanto para programadores que necessitam de maior flexibilidade (JAIN; DAVIS; MATVITSKY,

2024). Sua arquitetura *cloud-native* permite integração com diversos serviços e sistemas legados, tornando-a uma solução ideal para empresas em transformação digital. No entanto, sua rede de parceiros de implementação ainda é menor em comparação com concorrentes, o que pode dificultar sua adoção em larga escala (MATVITSKYY; DAVIS; JAIN, 2024).

#### 4. *OutSystems*

- a) **Abordar padrões de desenvolvimento centrados em dados e em processos:** *OutSystems* se destaca com ferramentas superiores para desenvolvimento focados na experiência do usuário, modelagem e gestão de dados, e desenvolvimento móvel, apoiados por investimentos líderes em inteligência artificial;
- b) **Fornecer ferramentas adequadas para diferentes perfis de desenvolvedores:** A plataforma é uma escolha proeminente para desenvolvedores profissionais, oferecendo acesso ao código gerado, que pode ser escaneado com ferramentas de teste de terceiros, facilitando o desenvolvimento de software de alto desempenho;
- c) **Atender aos requisitos específicos de infraestrutura, arquitetura e processo de desenvolvimento:** *OutSystems* oferece uma plataforma rica em funcionalidades para modernização *core low-code* e desenvolvimento rápido de padrões de aplicativos menores. No entanto, suas ferramentas para desenvolvedores de negócios são limitadas e seu gerenciamento de conteúdo é abaixo do ideal.

Dessa maneira, *OutSystems* é uma das plataformas *low-code* mais completas, oferecendo um conjunto avançado de ferramentas para desenvolvimento de aplicações empresariais e modernização de sistemas (MATVITSKYY; DAVIS; JAIN, 2024). Seu diferencial está na combinação de baixo código com capacidades de desenvolvimento profissional, permitindo que desenvolvedores tenham acesso ao código gerado para personalizações e otimizações avançadas (JAIN; DAVIS; MATVITSKYY, 2024). A plataforma também investe fortemente em inteligência artificial, fornecendo recursos para geração automática de código, otimização de processos e testes automatizados. Apesar de

ser uma das melhores opções para desenvolvedores técnicos, a *OutSystems* possui ferramentas limitadas para desenvolvedores cidadãos, o que pode restringir sua adoção em empresas que buscam democratizar o desenvolvimento de software (MATVITSKYY; DAVIS; JAIN, 2024).

## 5. *Appian*

- a) **Abordar padrões de desenvolvimento centrados em dados e em processos:** A plataforma *Appian* se destaca por sua forte abordagem na automação de processos e gestão de dados empresariais. Seu recurso *Process HQ* combina mineração de processos e capacidades analíticas, permitindo que as empresas realizem uma otimização contínua de seus fluxos de trabalho. Além disso, a *Appian* oferece um *Data Fabric*, que aprimora a virtualização e governança de dados, facilitando a integração com múltiplos sistemas e garantindo um alto nível de consistência e segurança dos dados. Essas funcionalidades são essenciais para empresas que buscam maior eficiência operacional e tomada de decisões baseadas em dados (JAIN; DAVIS; MATVITSKYY, 2024);
- b) **Fornecer ferramentas adequadas para diferentes perfis de desenvolvedores:** A *Appian* fornece um ambiente de desenvolvimento flexível, atendendo tanto desenvolvedores profissionais quanto desenvolvedores cidadãos. A plataforma disponibiliza quatro ambientes de desenvolvimento integrados (IDEs): *Appian Designer*, *Case Management Studio*, *Integration SDK* e *UI SDK*, garantindo suporte a diversos perfis de usuários. Além disso, a *Appian* integra-se com IDEs populares, como *Eclipse* e *IntelliJ IDEA*, permitindo que desenvolvedores experientes utilizem suas ferramentas preferidas. Para desenvolvedores cidadãos, a *Appian* oferece funcionalidades de geração de interfaces de usuário a partir de linguagem natural e documentos PDF, por meio do *Appian AI Copilot*, simplificando a criação de aplicações sem a necessidade de programação avançada (JAIN; DAVIS; MATVITSKYY, 2024);
- c) **Atender aos requisitos específicos de infraestrutura, arquitetura e processo de desenvolvimento:** A *Appian* permite uma im-

plantação flexível, oferecendo opções tanto para infraestrutura gerenciada pela própria plataforma quanto para infraestrutura gerenciada pelo cliente, com a maioria dos usuários optando pelo modelo devido à sua escalabilidade e menor necessidade de manutenção. A plataforma também fornece um amplo suporte para integração com ferramentas de CI/CD e monitoramento, permitindo que as empresas implementem fluxos de desenvolvimento contínuo e monitorem suas aplicações de forma eficiente (JAIN; DAVIS; MATVITSKYY, 2024).

Assim, a *Appian* é reconhecida por sua forte abordagem na automação de processos e modelagem de fluxos de trabalho, integrando ferramentas de mineração de processos e análise de dados para otimizar operações empresariais (MATVITSKYY; DAVIS; JAIN, 2024). Seu recurso *Process HQ* oferece análise contínua para melhoria de eficiência, enquanto o *Data Fabric* aprimora a virtualização e governança de dados, facilitando a integração com diversos sistemas (JAIN; DAVIS; MATVITSKYY, 2024). Além disso, a *Appian* fornece um ecossistema de desenvolvimento flexível, incluindo quatro IDEs: *Appian Designer*, *Case Management Studio*, *Integration SDK* e *UI SDK*. Sua parceria com a *Amazon Bedrock* fortalece suas capacidades em *GenAI*, permitindo geração automatizada de dados de teste e otimização de fluxos de trabalho. No entanto, a falta de suporte nativo a Kafka e *event-streaming* pode ser uma limitação para algumas organizações (MATVITSKYY; DAVIS; JAIN, 2024).

## 6. *PegaSystems*

- a) **Abordar padrões de desenvolvimento centrados em dados e em processos:** A plataforma *PegaSystems* é reconhecida por sua forte herança em automação de processos e otimização de fluxos de trabalho. Sua abordagem se baseia na inteligência artificial e na modelagem de processos, permitindo a criação de *workflows* altamente personalizados e automatizados. A ferramenta *Pega GenAI Blueprint* possibilita a geração automática de fluxos de trabalho, interfaces e modelos de dados a partir de comandos em linguagem natural, acelerando signifi-

cativamente o desenvolvimento de aplicações. Além disso, a plataforma oferece suporte nativo a mineração de processos (*process mining*), automação robótica de processos (RPA) e processamento inteligente de documentos (IDP), permitindo que as organizações capturem e analisem fluxos de dados em tempo real para otimizar suas operações (JAIN; DAVIS; MATVITSKYY, 2024);

- b) **Fornecer ferramentas adequadas para diferentes perfis de desenvolvedores:** A *PegaSystems* oferece um conjunto de ferramentas que atende tanto a desenvolvedores profissionais quanto a desenvolvedores cidadãos. Sua plataforma conta com diversos ambientes de desenvolvimento web, como *App Studio*, *Dev Studio* e *Robot Studio*, permitindo desde a construção visual de fluxos de trabalho até o desenvolvimento avançado de componentes personalizados e *bots* de automação. Através da integração com *IDEs* tradicionais e da disponibilização de *SDKs*, desenvolvedores podem estender funcionalidades utilizando suas linguagens e *frameworks* preferidos (JAIN; DAVIS; MATVITSKYY, 2024).

Além disso, a *PegaSystems* oferece ferramentas para automação de testes e geração de dados de teste baseada em IA, permitindo que os desenvolvedores validem suas aplicações de forma eficiente. A plataforma também conta com um forte suporte a controle de versão e colaboração simultânea, facilitando o desenvolvimento em equipe e garantindo que diferentes partes do sistema sejam gerenciadas e implantadas de forma integrada (JAIN; DAVIS; MATVITSKYY, 2024);

- c) **Atender aos requisitos específicos de infraestrutura, arquitetura e processo de desenvolvimento:** A *PegaSystems* oferece opções flexíveis de implantação, permitindo que seus clientes escolham entre infraestrutura gerenciada pela própria plataforma e infraestrutura gerenciada pelo cliente, com a maioria optando pelo modelo devido à facilidade de manutenção e escalabilidade.

Em termos de arquitetura, a *PegaSystems* fornece suporte robusto para desenvolvimento e integração de *APIs*, permitindo a conexão com diversas aplicações empresariais e bancos de dados. Seu recurso *Pega App*



*Factory* disponibiliza um painel centralizado para governança e monitoramento das aplicações desenvolvidas, garantindo que as organizações tenham total visibilidade sobre seu ambiente de desenvolvimento. Além disso, a plataforma conta com o *Pega Rule Security Analyzer*, uma ferramenta dedicada à análise e mitigação de falhas de segurança nas aplicações, assegurando conformidade e proteção dos dados.

No campo da inteligência artificial, a *PegaSystems* investe fortemente na utilização de *GenAI* para aprimorar a automação de processos e otimização de desempenho. Com o *Pega GenAI Blueprint* e o *Pega GenAI Autopilot*, a plataforma permite a criação de aplicações baseadas em IA, reduzindo significativamente o tempo necessário para modelagem de fluxos de trabalho e desenvolvimento de interfaces (JAIN; DAVIS; MATVITSKY, 2024).

Nesse modelo, a *PegaSystems* se destaca pela forte integração entre automação de processos, inteligência artificial e mineração de processos, permitindo a criação de *workflows* altamente personalizados e automatizados (MATVITSKY; DAVIS; JAIN, 2024). Seu diferencial está no *Pega GenAI Blueprint*, que possibilita a geração automatizada de fluxos de trabalho e interfaces através de linguagem natural, reduzindo significativamente o tempo de desenvolvimento (JAIN; DAVIS; MATVITSKY, 2024). A plataforma oferece um ecossistema robusto para desenvolvedores, com *IDEs* como *App Studio*, *Dev Studio* e *Robot Studio*, além de *SDKs* para integração com ferramentas externas. Em termos de infraestrutura e governança, a *Pega App Factory* fornece um painel centralizado para monitoramento, enquanto o *Pega Rule Security Analyzer* realiza auditorias de segurança em aplicações (MATVITSKY; DAVIS; JAIN, 2024). No entanto, alguns recursos avançados, como mineração de processos e suporte a *BPMN*, exigem licenciamento adicional, impactando os custos operacionais (JAIN; DAVIS; MATVITSKY, 2024).

Para Bratincevic e Koplowitz (2021), a *ServiceNow* é referência nas soluções de gerenciamento de produtos de software, por ter investido nas ferramentas *low-*

*code* e nos *workflows* - que são otimizados para a automação de tarefas. Além disso, destacam-se no quesito de gerenciamento de dados, por promover modelos de dados com diversas funcionalidades, dentre elas, estender objetos canônicos recorrendo às heranças, através de um simples clique.

Nota-se que a plataforma, por meio dessa abordagem, permite que os desenvolvedores incorporem elementos de dados diretamente dentro dos formulários, *workflows* e lógicas de negócio, com interações simples, ao invés de códigos personalizados e integrações via APIs ou outros *web services*. Assim, conforme ilustrado na Figura 4 e, de acordo com (KAGAN, 2021), a ServiceNow mostra-se uma empresa de popularidade alta na comunidade de desenvolvedores e fornece assistência no processo de design de serviços, ao promover instruções claras e consolidadas, a exemplo de aplicações desenvolvidas com foco em ambientes web, em que o processo técnico é centrado no usuário para que haja criação de valor.

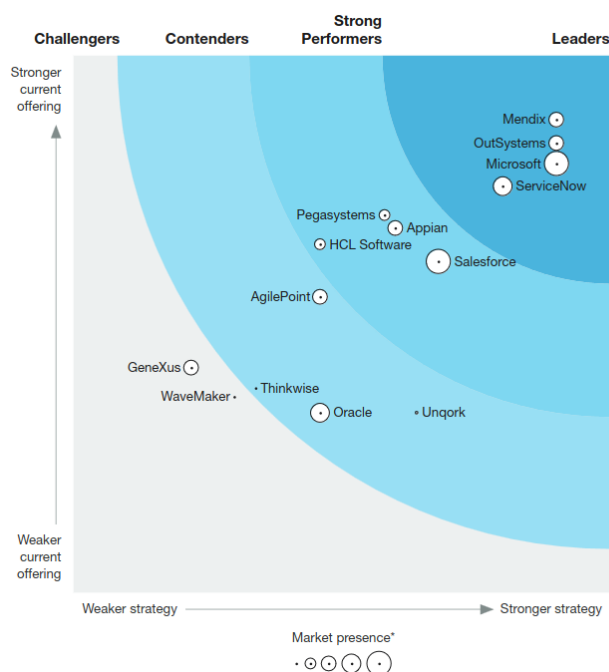


Figura 4 – Força da plataforma *ServiceNow* e sua capacidade de impactar a transformação digital juntamente com o design de serviços (BRATINCEVIC; KOPLOWITZ, 2021)

Em um estudo mais recente do *Gartner*, dos autores [Matvitskyy, Davis e Jain \(2024\)](#), nota-se que o quadrante das empresas provedoras de plataforma *low-code* líderes de mercado conta com a inclusão da empresa *Appian* e da empresa *PegaSystem*, conforme ilustrado na Figura 5:



Figura 5 – Quadrante de líderes - Atualização do quadrante “mágico” de empresas líderes de mercado([MATVITSKY; DAVIS; JAIN, 2024](#))

A necessidade de um rápido desenvolvimento e a constante iteração para com os produtos de software, de acordo com as ideias apresentadas por [Bratincevic e Koplowitz \(2021\)](#) e por [Matvitskyy, Davis e Jain \(2024\)](#), tornou o uso das plataformas *low-code/no-code* uma abordagem essencial no processo de transformação digital de diversas empresas, a destacar a *ServiceNow*, que destaca-se entre os fornecedores desses ambientes que têm procurado diferentes maneiras de aprimorar suas capacidades de atender segmentos mais especializados de casos de uso e diferentes perfis de desenvolvimento.

Neste trabalho, a *ServiceNow*, com foco no ambiente de desenvolvimento *App Engine Studio* é o objeto deste estudo.

## 2.7 Considerações Finais do Capítulo

Neste Capítulo, foi destacado como o *Design* de serviços, especialmente através do *codesign*, que integra os usuários como co-criadores no processo de design, garantindo que as soluções tecnológicas se alinhem às necessidades humanas. Foi abordado também a transformação digital, o desenvolvimento pelo usuário final (EUD), e as inovações em plataformas *low-code/no-code*. Foram discutidas como essas tecnologias reconfiguram processos, cultura e infraestrutura organizacionais para impulsionar a inovação e a eficiência operacional. Foi explorado como as plataformas *low-code/no-code* democratizam o desenvolvimento de software, facilitando a prototipagem rápida e a colaboração entre profissionais de TI e usuários finais, essenciais em mercados dinâmicos. No próximo Capítulo, apresenta-se a plataforma de desenvolvimento *low-code* ServiceNow, focando nos avanços significativos na cocriação de valor e inovação no desenvolvimento de serviços digitais.

## 3 Materiais e Métodos

### 3.1 Considerações Iniciais da Seção

Neste Capítulo apresenta-se o Plano Metodológico adotado, com um detalhamento do planejamento das 4 fases: Planejamento; Coleta de Dados, compostas pelos itens do Protocolo de Estudo de Caso; Análise de Dados; e Relatórios.

### 3.2 Planejamento de Pesquisa

Nesta fase de *planejamento* foi realizada a escolha do tema de pesquisa, a questão de pesquisa, e os objetivos, geral e específicos. Para alcançar o objetivo, foi definido um Plano Metodológico de Pesquisa.

A técnica de coleta de dados selecionada foi a de Estudo de Caso, uma vez que este trabalho visa examinar a aplicação do DS no contexto real de desenvolvimento de uma organização.

Para execução do Estudo de Caso foi definido a utilização do Protocolo proposto por [Brereton et al. \(2008\)](#). O Protocolo possui 11 itens, e a partir deles, os pesquisadores se orientam no planejamento e na execução da técnica de estudo de caso.

A Tabela 3 apresenta as etapas do Protocolo de [Brereton et al. \(2008\)](#), e a descrição de cada uma delas.

Tabela 3 – Protocolo de Estudo de Caso (BRERETON et al., 2008)

Etapa	Descrição
Background	a) Localizar estudos anteriores relacionados ao tópico b) Estabelecer a principal pergunta de pesquisa que será abordada neste estudo c) Identificar quaisquer perguntas de pesquisa adicionais que serão exploradas
Design	a) Determinar a escolha entre estudos de caso único ou múltiplos b) Detalhar o objeto do estudo, como um novo procedimento de teste ou um recurso recém-introduzido c) Identificar quaisquer proposições ou subquestões que se originem de cada questão de pesquisa
Seleção de Caso	a) Critério para seleção de caso
Procedimentos e papéis do estudo	a) Normas que regulam os processos de campo b) Responsabilidades dos membros da equipe de pesquisa do estudo de caso
Coleta de Dados	a) Identificação dos dados a serem coletados b) Estabelecimento de um plano de coleta de dados c) Definição de como será o armazenamento dos dados
Análise	a) Estabelecimento dos critérios para interpretar os resultados do estudo de caso b) Estabelecer quais dados serão utilizados para responder a questão da pesquisa c) Considerar as diversas possibilidades de resultados e identificar explicações alternativas d) A análise será realizada à medida que o estudo de caso avança
Validade do plano	a) Verificação geral para o projeto e o plano de coleta de dados b) Comprovação da validade interna, demonstrando a relação causal entre os resultados e a intervenção/tratamento c) Definição da validade externa, que identifica o âmbito de generalização dos resultados do estudo d) Garantia da validade de construto, mostrando que as medidas operacionais adequadas estão planejadas para os conceitos sob estudo
Limitações do Estudo	a) Especificação das questões de validade residual, incluindo potenciais conflitos de interesse que sejam inerentes ao problema, não decorrentes do plano
Relatórios	a) Identificação do público-alvo e a relação com estudos mais amplos
Cronograma	a) Fornecer estimativas de tempo para todas as etapas principais: planejamento, coleta de dados, análise de dados, geração de relatórios
Apêndices	a) Atualização contínua durante a condução do estudo, destacando quaisquer desvios em relação às etapas mencionadas anteriormente

### 3.3 Coleta de Dados

Como coleta, para a realização dos procedimentos 1, 2, 3, e 4, dado que os autores deste trabalho são colaboradores terceirizados e prestam serviços para a *Organização*, que é objeto deste estudo, foi acordado com representantes da área na *Organização*, a coleta de dados dela.

A aquisição dos dados da *Organização* foi realizada a partir de entrevistas com representantes-chave da *Organização*, da análise documental que foi disponibilizada aos pesquisadores, e do levantamento de dados dos processos de desenvolvimento à partir de *dashboards* da *Organização*.

Entre as atividades, foram realizados:

- Acesso e coleta de *dashboards*, para as métricas da *Organização* relacionadas com os objetivos deste trabalho;
- Acesso e coleta de documentos, entre eles, relatórios da *Organização* relacionados aos objetivos deste trabalho;
- Entrevistas junto aos gestores da *Organização*, buscando coletar e validar dados e *insights* relacionados com os objetivos deste trabalho.

### 3.4 Análise

A análise dos dados foi realizada concorrentemente com a coleta de dados, já que a abordagem possibilita que novas interpretações sejam encontradas durante a análise.

### 3.5 Relatório

O Relatório desta pesquisa se materializa como esta monografia de TCC.

### 3.6 Atividades do Protocolo de Estudo de Caso

Seguindo o Protocolo de Estudo de Caso adotado, nesta Seção, apresenta-se uma descrição dos itens.

#### 3.6.1 *Background*

Na etapa de *Background* os pesquisadores exploraram a *Pesquisa Bibliográfica* e a *Pesquisa Documental* relacionadas ao tópico investigado. Isso envolveu uma revisão da literatura para entender o contexto, as descobertas e as lacunas já identificadas no estado da arte. Nessa etapa a principal pergunta de pesquisa foi formulada, assim como a identificação de outras questões adicionais a partir da revisão da literatura.

### 3.6.2 Design

O Design deste estudo de caso é a escolha de um caso único. Isso é, uma organização que emprega uma plataforma de desenvolvimento *low-code* para desenvolver as soluções de software.

No caso, a plataforma *ServiceNow* foi selecionada para uma análise mais detalhada, no qual foi estudado o ambiente de desenvolvimento APS. O objetivo foi investigar como ele, com o emprego de DS, tem apoiado a TD na *Organização*.

### 3.6.3 Seleção de Caso

Quanto a seleção de caso, dado que este trabalho visa examinar a aplicação do DS no contexto real de desenvolvimento de uma organização, o objeto de seleção de caso foi uma organização do setor financeiro, denominada ficticiamente neste trabalho de *Organização*, que faz uso da plataforma *Servicenow* (Capítulo 4).

Por questões de privacidade, a *Organização* não será identificada. Contudo, no contexto de Transformação Digital, apresenta-se que a *Organização* possui 03 (três) pilares de fundamentação, que possibilitam a análise dos dados do impacto do emprego do *low-code*, promovendo, assim, o atingimento dos objetivos específicos desta da pesquisa. Os pilares podem ser descritos como:

- Promover a Experiência Digital e a Inovação;
- Promover a Integração dos Processos e das Soluções;
- Promover a Cultura Digital para a inovação e para a equidade.

A *Organização* emprega o ambiente de desenvolvimento APS da plataforma *low-code* possibilita a criação rápida e eficiente de aplicativos empresariais, promovendo a colaboração entre desenvolvedores técnicos e não técnicos e a cocriação com usuários finais para garantir soluções alinhadas às suas necessidades e expectativas.

O APS facilita o design, a construção, o teste e a implementação de aplicativos sem a necessidade de extensa codificação, promovendo assim uma maior colaboração entre desenvolvedores técnicos e não técnicos.



Além disso, inclui ferramentas intuitivas para modelagem de dados, design de interface do usuário, implementação de lógica de negócios, integração de serviços externos, e testes e implementação de aplicativos. Este ambiente é ideal para as diversas abordagens que cercam a TD, pois permite que múltiplos *stakeholders*, incluindo usuários finais, participem ativamente no processo de desenvolvimento, garantindo que as soluções criadas sejam alinhadas com as necessidades e expectativas dos usuários.

Assim, como justificativa da escolha desta *Organização* como objeto de seleção de caso, tem-se:

1. Os 02 (dois) pesquisadores, autores deste trabalho, trabalham em empresas que prestam serviços de desenvolvimento para esta *Organização*, como terceirizados *low-code*, possibilitando o acesso aos dados.
2. A *Organização* emprega o *low-code* juntamente com os princípios da abordagem de DS.

### 3.6.4 Procedimentos e papéis do estudo de caso

Quanto aos procedimentos deste trabalho, estes estão alinhados aos objetivos específicos do trabalho, compondo assim, os 04 (quatro) procedimentos. São eles:

1. Investigar o quanto o *low-code* contribui para a mitigação do *Shadow IT* na *Organização*, e consequentemente, para a TD;
2. Investigar o emprego do *low-code* sob a perspectiva dos pilares de TD da *Organização*;
3. Analisar o quanto *low-code* tem acelerado o desenvolvimento de soluções digitais e de novas funcionalidades para a Integração de Processos e Soluções da *Organização*;
4. Explorar como os princípios do *Design de Serviços* foram integrados nas soluções desenvolvidas com a abordagem *low-code/no-code* na evolução do ambiente organizacional.

Quanto aos papéis, os autores, pesquisadores deste trabalho, têm vínculo profissional com as empresas que prestam serviços à *Organização*. Ambos são desenvolvedores plenos e pertencem ao mesmo time na empresa.

### 3.6.5 Coleta de Dados

A coleta de dados para esta monografia engloba tanto a pesquisa bibliográfica quanto a pesquisa documental.

A coleta de dados da *Organização* teve como foco a investigação do impacto do emprego do *low-code* em uma empresa, no contexto de DS, visando à TD.

A coleta é resultante de entrevistas com os membros da *Organização*. Foram realizadas entrevistas com 03 (três) diferentes perfis, sendo:

- Gerente de Nível Estratégico;
- Gerente de Time de Desenvolvimento;
- *Citizen Developer*.

Além da análise documental, focada principalmente em:

- Acesso e coleta de documentos, entre eles, relatórios da *Organização*;
- Acesso aos *dashboards*, para as métricas e *insights* organizacionais.

### 3.6.6 Análise

A partir da coleta, o objetivo da análise é abstrair conclusões sólidas e embasadas a partir das informações coletadas, mantendo uma cadeia clara de evidências que sustente as conclusões alcançadas.

O objetivo foi investigar o quanto a abordagem *Low-code*, utilizando o ambiente de desenvolvimento App Engine Studio da ServiceNow, empregada no contexto de DS, tem apoiado a estratégia de TD da *Organização*.

### 3.6.7 Validade do plano

Com a Coleta de dados, a análise e relato desses foram apresentados para validação a alguns funcionários, de papéis diversos, da *Organização*.

### 3.6.8 Limitações do Estudo

Trata-se de caso único. Os dados levantados se baseiam na *Organização*. Durante o desenvolvimento deste trabalho, foram especificadas questões da validade residual deste estudo e suas limitações.

### 3.6.9 Relatório

Esta fase é caracterizada pela monografia deste trabalho de dissertação.

### 3.6.10 Cronograma

O cronograma desta proposta é também o cronograma da pesquisa, apresentado na Figura 6.

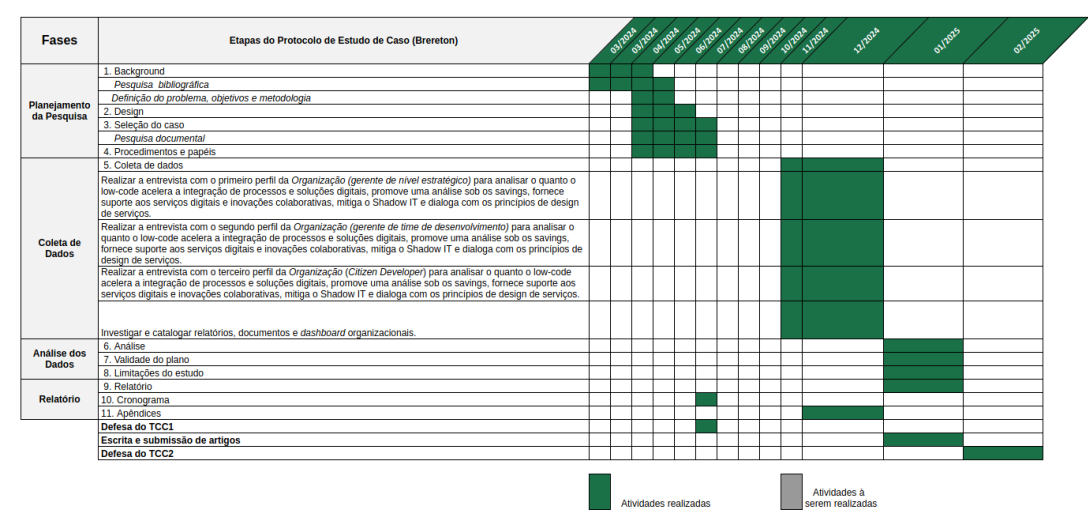
As atividades compreenderam os procedimentos da pesquisa, considerando as 04 (quatro) fases do Planejamento Metodológico adotado e os itens do Protocolo de Estudo de Caso adotado. Além das atividades do Trabalho de Conclusão de Curso - TCC, como o TCC1 e o TCC2, assim como a defesa e a iniciação da escrita de um artigo para submissão a um evento.

### 3.6.11 Apêndices

Os documentos produzidos pelos autores deste trabalho, como as cartas de agradecimento e apresentação e os documentos com as diretrizes das entrevistas, originadas do planejamento e execução do *workshop* constam como apêndices:

- Apêndice *A* - *Roteiro das Entrevistas*;
- Apêndice *B* - *Carta de Apresentação*;
- Apêndice *C* - *Carta de Agradecimento*.

Figura 6 – Cronograma do Projeto de Pesquisa



3.7 Considerações Finais do Seção

Neste Capítulo apresentou-se o Plano Metodológico adotado neste trabalho. No capítulo seguinte, apresenta-se a plataforma *ServiceNow*, com um foco específico em sua ferramenta de desenvolvimento *low-code/no-code*, o *App Engine Studio*.

## 4 *Plataforma ServiceNow*

### 4.1 Considerações Iniciais do Capítulo

Este Capítulo tem como objetivo apresentar a plataforma *ServiceNow*, com um foco específico em sua ferramenta de desenvolvimento *low-code/no-code*, o *App Engine Studio*. Inicia-se com uma visão geral da *ServiceNow*, explorando sua história, missão e visão. Em seguida, apresentam-se as funcionalidades e os benefícios do *App Engine Studio*, destacando como esse incorpora princípios de métodos ágeis e busca promover a colaboração entre diferentes papéis no desenvolvimento de aplicativos. Finaliza-se com a apresentação da integração do usuário como componente central desta plataforma, possibilitando que as soluções possam ser desenvolvidas pelo próprio usuário, e até mesmo, apoiar no emprego de abordagens que sejam centradas no usuário.

### 4.2 Visão Geral da *ServiceNow*

A *ServiceNow* é uma plataforma de automação de fluxos de trabalho baseada em nuvem que promove uma nova forma de como as organizações gerenciam seus processos e operações ([ServiceNow, 2024b](#)).

Fundada em 2004 por Fred Luddy, a empresa se estabeleceu como um líder no mercado de gestão de serviços de TI (*IT Service Management* - ITSM) e, desde então, expandiu suas ofertas para incluir operações de TI, atendimento ao cliente, recursos humanos, entre outros ([RAMIREZ, 2023](#)).

Com uma presença global, a *ServiceNow* atende a milhares de clientes em diversas indústrias, incluindo finanças, saúde, educação e governo. Algumas das empresas que empregam os serviços da *ServiceNow* para otimizar suas operações e promover a transformação digital são a Coca-Cola, *Amazon*, *Siemens*, *Microsoft*, *GE Healthcare*, *Nike* e *Novartis*. Essas organizações utilizam a plataforma *ServiceNow* para gerenciar uma ampla gama de processos, desde a gestão de serviços de

TI até a automação de fluxos de trabalho empresariais complexos, demonstrando a versatilidade e eficácia das soluções oferecidas (WILLSON, 2023).

A *ServiceNow* tem sido um facilitador na jornada de transformação digital de muitas empresas. Suas soluções permitem que as organizações automatizem tarefas repetitivas, melhorem a eficiência operacional e ofereçam melhores serviços aos clientes e funcionários. A capacidade da *ServiceNow* de integrar-se perfeitamente com outras ferramentas e sistemas empresariais facilita a adoção de novas tecnologias e processos, ajudando as empresas a se adaptarem rapidamente às mudanças do mercado e às demandas dos clientes (ServiceNow, 2024b).

Nesse contexto, o *App Engine Studio* se destaca como um ambiente de desenvolvimento essencial dentro do ecossistema *ServiceNow*, oferecendo uma plataforma de desenvolvimento *low-code* que empodera os usuários a criar aplicativos personalizados de maneira rápida e eficiente. No próximo segmento, serão explorados em detalhes como o *App Engine Studio* contribui para a transformação digital, apoiado pelo design de serviços nas organizações modernas (ServiceNow, 2024b).

### 4.3 *App Engine Studio*

O ambiente de desenvolvimento, denominado *App Engine Studio* (APS) é um ambiente *low-code* da *ServiceNow* que possibilita a criação rápida e eficiente de aplicativos empresariais personalizados. No APS, diferentes papéis desempenham funções cruciais no desenvolvimento e implantação de aplicativos. O APS suporta a colaboração entre usuários, reconhecidos na plataforma como desenvolvedores cidadãos, desenvolvedores profissionais e administradores de TI, permitindo que todos contribuam de acordo com suas habilidades e responsabilidades.

Com uma interface intuitiva, o APS facilita a criação de aplicativos empresariais personalizados, promovendo a eficiência e agilidade na transformação digital. Além disso, a plataforma integra ferramentas para automação de fluxos de trabalho, segurança e governança, garantindo que as soluções desenvolvidas sejam robustas e alinhadas às necessidades dos usuários (ServiceNow, 2024a).

### 4.3.1 Papéis no Desenvolvimento com *App Engine Studio*

No desenvolvimento de aplicações de software utilizando o APS, diferentes papéis são atribuídos para garantir que as aplicações sejam construídas de forma eficiente e colaborativa, ao envolver diferentes perfis. Cada papel desempenha uma função específica que contribui para o sucesso do projeto, desde a configuração inicial do ambiente até a entrega final das aplicações. Tem-se assim, uma lista em tópicos com uma visão geral dos principais papéis envolvidos no desenvolvimento com o APS ([ServiceNow, 2024a](#)):

#### 1. *Desenvolvedor Cidadão (Citizen Developer)*

##### a) **Responsabilidades:**

- i. Criar Aplicações utilizando ferramentas *low-code/no-code* sem a necessidade de conhecimentos avançados de programação;
- ii. Prototipar rapidamente ideias e soluções que atendam as necessidades específicas do negócio.

##### b) **Impacto:**

- i. Democratizar o desenvolvimento de software, permitindo que mais membros da organização contribuam para a inovação e a transformação digital de maneira colaborativa;
- ii. Facilitar a rápida iteração e implementação de melhorias baseadas no *feedback* contínuo.

#### 2. *Desenvolvedor Profissional (Professional Developer)*

##### a) **Responsabilidades:**

- i. Desenvolver soluções complexas e customizadas, integrando a plataforma *ServiceNow* com outros sistemas e serviços;
- ii. Garantir que as soluções atendam aos requisitos técnicos e de segurança.

##### b) **Impacto:**

- i. Complementar o trabalho dos desenvolvedores cidadãos, fornecendo suporte técnico avançado e garantindo a escalabilidade e robustez das soluções;
- ii. Atuar como um recurso para resolver problemas técnicos que excedam as capacidades das ferramentas *low-code/no-code*.

### 3. *Analista de Negócios (Business Analyst)*

#### a) **Responsabilidades:**

- i. Coletar e documentar requisitos de negócios e traduzi-los em especificações técnicas para os desenvolvedores;
- ii. Facilitar a comunicação entre as partes interessadas e a equipe de desenvolvimento, garantindo que as soluções atendam às necessidades do negócio.

#### b) **Impacto:**

- i. Assegurar que as soluções desenvolvidas estão alinhadas com os objetivos estratégicos da organização;
- ii. Promover uma compreensão clara dos requisitos e expectativas, melhorando a qualidade e relevância das soluções desenvolvidas.

### 4. *Especialista em Experiência do Usuário (UX Specialist)*

#### a) **Responsabilidades:**

- i. Projetar interfaces intuitivas e amigáveis para o usuário;
- ii. Conduzir testes de usabilidade e coletar *feedback* dos usuários para melhorar a interface.

#### b) **Impacto:**

- i. Melhorar a adoção e satisfação do usuário final, assegurando que as soluções são fáceis de usar e eficazes;
- ii. Garantir que o design do serviço seja centrado no usuário, promovendo uma experiência de alta qualidade.

### 5. *Proprietário do Produto (Product Owner)*



**a) Responsabilidades:**

- i. Definir a visão do produto e priorizar os itens do *backlog*, tomando decisões rápidas sobre o desenvolvimento e o escopo do produto;
- ii. Garantir que o produto final atenda às necessidades dos usuários e aos objetivos do negócio.

**b) Impacto:**

- i. Fornecer direção estratégica clara para o desenvolvimento do produto, alinhando o trabalho da equipe com as prioridades de negócios;
- ii. Facilitar a entrega de valor contínuo ao garantir que o desenvolvimento esteja focado nas funcionalidades mais importantes e de maior impacto.

A colaboração entre esses papéis é fundamental para o sucesso do design de serviços utilizando uma abordagem de *low-code/no-code* na plataforma de desenvolvimento *ServiceNow*. Essa colaboração assegura que todas as perspectivas necessárias sejam consideradas, desde a visão estratégica até a experiência do usuário final. A integração contínua de *feedback* de todas as partes interessadas permite a criação de soluções mais adaptadas às necessidades reais da organização. O design de serviços centrado no usuário, alimentado pela colaboração e iteração contínua, levando em consideração o holismo, garante que as soluções sejam não apenas tecnicamente sólidas, mas também altamente eficazes e bem aceitas pelos usuários finais (ServiceNow, 2024a), (BUCAIONI; CICHETTI; CICOZZI, 2022).

Esses cinco papéis são essenciais para maximizar o impacto abordagem de *low-code/no-code* na transformação digital dentro de um framework de design de serviços. Eles representam a colaboração necessária entre usuários com diferentes perfis, a garantir, dessa forma, que as soluções desenvolvidas sejam robustas, escaláveis e alinhadas com as metas estratégicas da organização e dos usuários finais (BUCAIONI; CICHETTI; CICOZZI, 2022), (ServiceNow, 2024a).

### 4.3.2 Princípios da Metodologia Ágil no APS

Ao facilitar iterações rápidas através de suas funcionalidades de desenvolvimento ágil, com o uso de ciclos de desenvolvimento curtos, contínuos e colaborativos, o APS da ServiceNow integra profundamente os métodos ágeis em seu núcleo, proporcionando uma plataforma eficaz para o desenvolvimento e a implantação de aplicativos empresariais. Desenvolvedores podem rapidamente prototipar, testar e ajustar aplicativos utilizando a interface do *App Engine Studio* e suas diversas ferramentas internas. Dito isso, entre alguns dos princípios de metodologia ágil que compõem o APS estão ([ServiceNow, 2024a](#)):

- **Iteração Rápida:** Por exemplo, o Flow Designer, permite a criação de fluxos de trabalho complexos em questão de minutos, sem a necessidade de codificação extensa. Isso permite que as equipes de desenvolvimento respondam rapidamente a mudanças nos requisitos e prioridades, mantendo um ritmo de entrega constante e ágil;
- **Feedback Contínuo:** A plataforma integra ferramentas robustas para a coleta e análise de feedback contínuo dos usuários finais. Utilizando o UI Builder, as equipes podem construir interfaces de usuário intuitivas e lançar versões preliminares para obter feedback dos usuários em tempo real. A funcionalidade de Mobile App Builder permite a criação de aplicativos móveis que podem ser testados diretamente pelos usuários, proporcionando insights valiosos sobre a experiência e usabilidade do aplicativo. Esse feedback é então incorporado nas próximas iterações, garantindo que o desenvolvimento esteja sempre alinhado com as necessidades e expectativas dos usuários;
- **Colaboração entre Equipes:** O App Engine Studio promove uma forte colaboração entre diferentes papéis dentro da organização. Desenvolvedores cidadãos podem utilizar o Catalog Builder para criar rapidamente novos serviços e produtos, enquanto os desenvolvedores profissionais podem usar o Integration Hub para conectar esses serviços a sistemas e dados existentes. Administradores de TI utilizam ferramentas de controle de acesso para garantir que todas as alterações sejam seguras e conformes. Essa integração de

funções permite que as equipes trabalhem juntas de maneira mais eficiente, com cada membro contribuindo de acordo com suas especialidades;

- **Automação de Fluxo de Trabalho:** A automação é um elemento chave no App Engine Studio, facilitando o desenvolvimento ágil. O Flow Designer e o Process Automation Designer permitem que as equipes automatizem processos complexos com facilidade. Por exemplo, um processo de aprovação de solicitações pode ser automatizado para enviar notificações e coletar aprovações em várias etapas, reduzindo o tempo de espera e erros manuais. Essas ferramentas permitem que as equipes se concentrem em tarefas de maior valor, melhorando a eficiência geral do desenvolvimento;
- **Segurança e Governança:** Apesar da agilidade proporcionada pelo App Engine Studio, a segurança e a governança não são comprometidas. A plataforma inclui ferramentas integradas de Access Control que garantem que os aplicativos sejam desenvolvidos e implantados de forma segura. Por exemplo, administradores podem definir permissões detalhadas para diferentes usuários e funções, garantindo que apenas indivíduos autorizados possam fazer alterações críticas. Além disso, o sistema de monitoramento contínuo da ServiceNow permite que as equipes identifiquem e respondam rapidamente a potenciais ameaças de segurança.

### 4.3.3 Processo de Desenvolvimento do APS

O processo de trabalho para criar aplicações no APS, da *ServiceNow*, é projetado para ser intuitivo e colaborativo ([ServiceNow, 2024a](#)), permitindo que desenvolvedores cidadãos e profissionais contribuam de maneira eficaz para o desenvolvimento de soluções.

O processo envolve 09 etapas, desde a submissão inicial da ideia até a implantação final no ambiente de produção, com o propósito de assegurar que as aplicações sejam desenvolvidas com qualidade e alinhadas às necessidades do negócio ([ServiceNow, 2024a](#)).

Apresenta-se um breve relato de cada uma das etapas do processo, além de representá-las na Figura 8:

Figura 7 – Processo de criação de aplicações usando o APS (ServiceNow, 2024a)

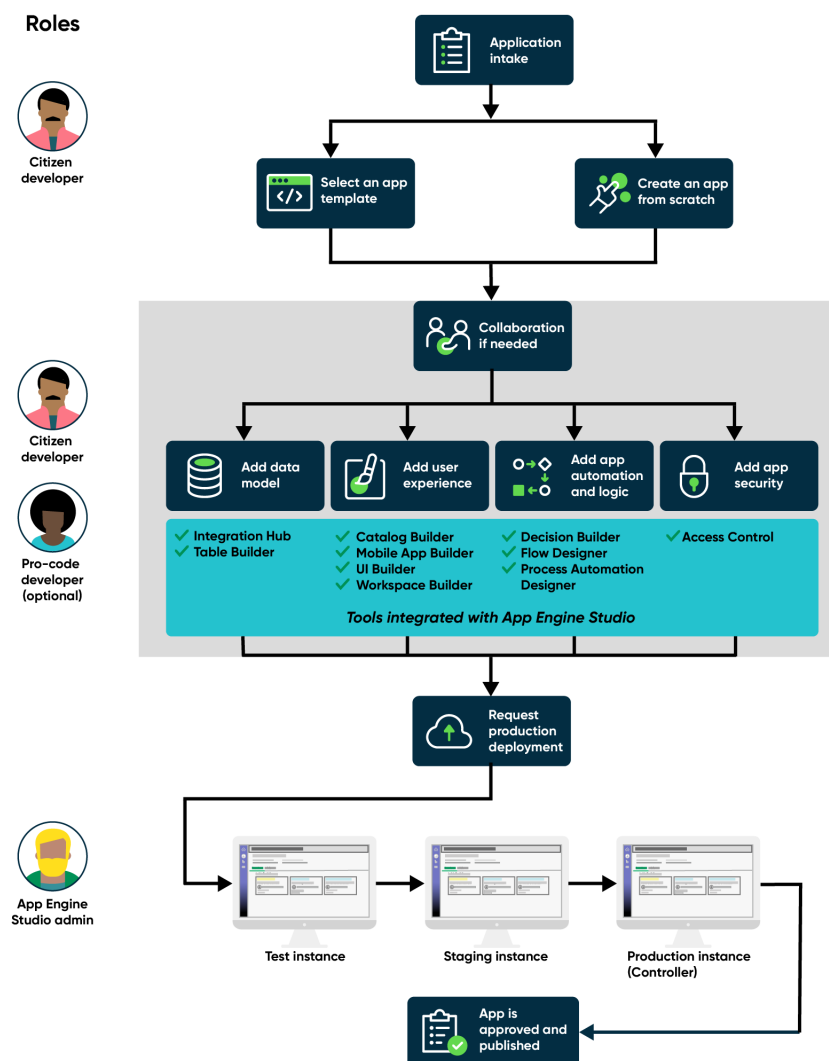


Figura 8 – Processo de criação de aplicações usando o APS (ServiceNow, 2024a)

1. **Submissão do Plano do Aplicativo** - Como desenvolvedor cidadão, o primeiro passo é enviar seu plano para uma aplicação através do processo de admissão de aplicações no APS. Esta submissão deve incluir uma descrição clara da ideia do aplicativo, seus objetivos e os benefícios esperados;
2. **Revisão da Solicitação pelo Administrador APS** - O administrador do APS revisa a solicitação de admissão. Se a solicitação for rejeitada, ela será encerrada. No entanto, o desenvolvedor pode atualizar sua ideia com base no *feedback* recebido e enviar uma nova solicitação. Esse processo iterativo permite refinamento contínuo das ideias propostas;
3. **Decisão sobre o Uso de Modelos** - Após a aprovação da solicitação, o desenvolvedor deve decidir se deseja criar o aplicativo utilizando um modelo predefinido ou desenvolver a aplicação do zero. O APS oferece diversos modelos que podem economizar tempo e esforço, proporcionando uma estrutura básica que pode ser customizada conforme necessário;
4. **Avaliação dos Recursos Disponíveis** - O desenvolvedor deve avaliar se possui os recursos necessários para criar a aplicação por conta própria. Se precisar de ajuda, pode solicitar colaboração, ou o administrador APS pode atribuir um desenvolvedor colaborador para auxiliar no projeto;
5. **Criação do Aplicativo com Modelos** - Se um modelo for utilizado, o sistema cria automaticamente uma aplicação base, incluindo um modelo de dados básico, experiências do usuário, fluxos de trabalho automatizados e funções de segurança. Esses elementos podem ser personalizados para atender às necessidades específicas do aplicativo;
6. **Desenvolvimento do Aplicativo do Zero** - Se o desenvolvedor optar por criar a aplicação do zero, ele deverá definir manualmente os quatro elementos fundamentais: modelo de dados, experiências do usuário, fluxos de trabalho e funções de segurança. Este processo oferece maior flexibilidade, permitindo a criação de soluções altamente customizadas;
7. **Monitoramento do Desenvolvimento pelo Administrador APS** - O administrador APS pode monitorar o progresso do desenvolvimento da apli-

cação em todas as etapas utilizando a Central de Gestão do APS. Esse monitoramento ajuda a equilibrar as cargas de trabalho e garantir a qualidade das aplicações em desenvolvimento;

8. **Implantação no Pipeline para Teste e Preparação** - Uma vez concluída, a aplicação é submetida para teste, preparação e implantação final no pipeline. O administrador utiliza a Central de Gestão do APS para gerenciar essa transição, assegurando que a aplicação esteja pronta para o ambiente de produção;
9. **Aprovação e Implantação Final** - Após a aprovação final, a aplicação é implantada no ambiente de produção, tornando-se disponível para os usuários finais. Este processo garante que a aplicação tenha passado por todas as etapas necessárias de desenvolvimento e testes, assegurando sua qualidade e funcionalidade.

Dessa maneira, o processo de trabalho no *App Engine Studio* é projetado para ser colaborativo e eficiente, integrando desenvolvedores de diferentes níveis de habilidade e garantindo que as aplicações sejam desenvolvidas de acordo com as melhores práticas e necessidades do negócio. A combinação de ferramentas *low-code/no-code*, modelos pré-definidos e monitoramento contínuo permite a criação de soluções inovadoras e adaptáveis, promovendo a transformação digital dentro das organizações (ServiceNow, 2024a).

## 4.4 Considerações Finais do Capítulo

Este Capítulo destacou a plataforma *ServiceNow* e seu ambiente de desenvolvimento *App Engine Studio*, uma ferramenta *low-code* que facilita o desenvolvimento de aplicativos empresariais de forma rápida e colaborativa. A *ServiceNow*, com sua abordagem centrada no usuário e métodos ágeis, busca promover a automação e a eficiência operacional, permitindo que organizações adaptem-se rapidamente às demandas do mercado. No contexto do *App Engine Studio*, foram delineados os papéis fundamentais no desenvolvimento colaborativo, como

desenvolvedor cidadão, desenvolvedor profissional, analista de negócios, especialista em UX e proprietário do produto, todos essenciais para maximizar o impacto do *low-code* na transformação digital. A plataforma busca, não só empoderar diversos perfis dentro da organização a contribuírem para a inovação, mas também garantir que as soluções sejam robustas, seguras e alinhadas às necessidades dos usuários finais, reforçando o design de serviços centrado no usuário. E apresenta o *App Engine Studio* como um exemplo de *low-code*, que apoiado por uma colaboração efetiva, pode impulsionar a transformação digital e criar valor contínuo para as organizações.

## 5 Impacto do *low-code* na *Organização*

### 5.1 Considerações Iniciais do Capítulo

Neste Capítulo são apresentados os resultados da investigação dos dados da aplicação da abordagem *low-code* na *Organização* estudada, utilizando o *App Engine Studio*, da *ServiceNow*, como ferramenta central. Inicialmente, são descritos os procedimentos adotados para coleta de dados, detalhando as etapas do estudo de caso e a relação entre os dados obtidos e os objetivos específicos definidos nesta pesquisa. Em seguida, a partir das percepções dos envolvidos na *Organização*, levantados por entrevistas realizadas, complementando a investigação deste trabalho. Em seguida, o compêndio dos resultados são apresentado aos envolvidos da *Organização* para validação e refinamento, com foco nas principais contribuições da abordagem *low-code* para o aprimoramento dos processos organizacionais, para a mitigação de desafios como o *Shadow IT*, e, para a promoção da TD. O objetivo foi consolidar as evidências do impacto do *low-code* na *Organização*, alinhando-se às premissas estabelecidas nos Capítulos anteriores.

### 5.2 Procedimentos de Coleta de Dados

Neste tópico, são apresentados os métodos utilizados para a Coleta de Dados no Estudo de Caso. A coleta, para a realização dos procedimentos 1, 2, 3 e 4, do Capítulo 3, dado que os autores deste trabalho são colaboradores e prestam serviços para a *Organização*, foi realizada por meio de entrevistas semiestruturadas com diferentes perfis organizacionais, análise de relatórios, acesso a dados da organizacionais e pesquisas sobre o ambiente organizacional, todos fornecidos pela *Organização*. Essa abordagem buscou garantir a abrangência e a qualidade das informações coletadas, alinhando-se aos objetivos definidos no planejamento da pesquisa.

Entre as atividades, foram realizadas:



- Acesso e coleta de *dashboards* e de diferentes documentos, entre eles, relatórios da *Organização* relacionados aos objetivos deste trabalho, para as métricas da relacionadas com os objetivos deste trabalho;
- Entrevistas junto aos gestores da *Organização*, buscando coletar e validar dados e *insights* relacionados com os objetivos deste trabalho.

A seguir, cada uma é detalhada em subseções específicas.

### 5.2.1 Entrevistas com Representantes da Organização

Foram realizadas 03 (três) entrevistas foram realizadas remotamente, com o emprego do aplicativo *Microsoft Teams*, com representantes de diferentes níveis hierárquicos, envolvendo os perfis de *gerente de nível estratégico* da plataforma, *Citizen Developer* e *gerente de equipe de desenvolvimento*.

As entrevistas foram conduzidas sob anonimato, promovendo a liberdade dos entrevistados em compartilhar suas experiências e percepções sem receios. Cada perfil entrevistado trouxe contribuições significativas, considerando suas rotinas e contextos distintos.

O gerente de nível estratégico da plataforma destacou seu papel na definição do *roadmap* de soluções e na gestão de equipes especializadas, com foco na entrega eficiente de atualizações e no suporte aos desenvolvedores cidadãos. O *Citizen Developer*, com formação inicial em Engenharia Agrônômica, compartilhou sua trajetória de transição para a área tecnológica, abordando a superação de desafios iniciais e a exploração do potencial do *low-code* na resolução de problemas administrativos e operacionais. Já o gerente de time de desenvolvimento, com atuação voltada à governança de TI e à integração de processos, ressaltou os desafios enfrentados na adaptação de modelos de governança e na mitigação de práticas de *Shadow IT*, promovendo o TD por meio de soluções inovadoras.

As entrevistas foram realizadas utilizando um formato semiestruturado, buscando explorar as perspectivas dos entrevistados sobre a adoção e impacto do *low-code* na *Organização*, e com isso revelar *insights* significativos sobre a integração dessa tecnologia nos processos organizacionais. Buscou-se identificar avanços,

desafios e oportunidades relacionadas à TD, buscando avaliar como o uso do *App Engine Studio* contribuiu para a transformação digital da *Organização*, ao mesmo tempo em que investigou desafios enfrentados pelos diferentes perfis e as oportunidades de melhoria percebidas pelos entrevistados.

Os tópicos abordados nas entrevistas incluíram:

- Impacto do *low-code* na aceleração da entrega de soluções e na mitigação de *Shadow IT*;
- Adoção do *App Engine Studio*, com destaque para a integração de diferentes perfis de desenvolvedores dentro da plataforma;
- Desafios da abordagem tradicional, em relação às políticas organizacionais, em contraste com a flexibilidade e governança proporcionada pela plataforma *low-code*;
- Contextos para planejamento estratégico da plataforma *low-code* a curto e a longo prazo dentro da *Organização*;
- Entendimento do contexto de desenvolvimento de soluções de software feitas dentro da plataforma;
- Entendimento de como a plataforma alinha-se com os pilares, estabelecidos pela *Organização*, para a promoção eficaz da TD;
- Resultados observados, como melhorias em eficiência, colaboração interna e alinhamento estratégico.

Esses temas foram explorados com base em um roteiro semiestruturado, apresentado no Apêndice (A), garantindo a coleta de percepções alinhadas aos objetivos da pesquisa, possibilitando também a flexibilidade para *insights* adicionais.

Nos próximos subtópicos, cada um desses aspectos é detalhado e discutido, à luz dos depoimentos e das referências conceituais abordadas ao longo da monografia, proporcionando uma compreensão mais profunda sobre o impacto do *low-code* no apoio a estratégia de TD da *Organização*. Além disso, ao final de

cada subtópico, apresenta-se a visão e as conclusões retiradas das entrevistas e dos cruzamentos realizados com a base teórica presente neste trabalho.

#### 5.2.1.1 *Shadow IT* e o contexto do low-code na Organização

Conforme contextualizado na Seção 1.2, do Capítulo 1, o surgimento de novas tecnologias, como a computação em nuvem, promoveu alterações no modelo de negócio de diversas empresas, tornando-as, em sua essência, empresas que prestam serviços digitais. A TD torna-se, então, um ponto central para a evolução e progresso delas. Todavia, o uso e surgimento de novas tecnologias, tornou mais difícil para áreas de fora do corpo diretivo de tecnologia desenvolver suas aplicações. Com isso, o *Shadow IT* aparece como um impasse para a política de governança e de gerenciamento de ativos de TI (GASPAR; MONTEIRO, 2018).

Nesse contexto, conforme informado na entrevista com o gerente de equipe da Organização, muitos times demandavam soluções de TI por meios não oficiais, devido ao fato de as alternativas tradicionais serem burocráticas e custosas. Isso também é ilustrado por Zimmermann e Rentrop (2014), ao definir essa dificuldade como alto custo de transação dentro das instituições.

Ainda na perspectiva desse colaborador da Organização, com o low-code, esse cenário mudou. Passou a ser oferecido um escopo onde os usuários podem desenvolver suas aplicações sem depender de infraestrutura adicional, como servidores ou bancos de dados. Assim, elimina-se a necessidade de ferramentas caseiras, como processos não automatizados e que careciam de controle e de segurança. Além disso, o low-code permite que tanto a TI quanto os times de negócio otimizem recursos e trabalhem com mais autonomia, sem perder governança. Isso diminui drasticamente a incidência de *Shadow IT*.

Além disso, para ele, o low-code diminui o custo de transação ao eliminar a necessidade de recursos adicionais, como servidores, bancos de dados ou camadas complexas de infraestrutura. Com plataformas como o ServiceNow, tem-se uma solução na nuvem pronta para receber o desenvolvimento de aplicações que suprem as necessidades das áreas de negócio. Isso tudo age ao reduzir a dependência da TI, o que gera benefícios tanto para os usuários, que ganham agilidade, quanto para

a *Organização*, que economiza ao não precisar alocar ou gerenciar esses recursos, por contar com a infraestrutura de governança da plataforma.

Outro ponto relevante levantado durante a entrevista com esse colaborador, tange sobre a existência de *Shadow IT* dentro da plataforma de *low-code* da *Organização*, por conta da facilidade e agilidade com que aplicações são criadas. Nesse contexto, para ele, existe esse risco, porém começaram a gerenciar isso, ainda que em processo de amadurecimento, com o auxílio da governança de TI possibilitada pela plataforma.

Por fim, na perspectiva do gerente de nível estratégico da *Organização*, a concepção da plataforma foi criada para combater o *Shadow IT*. Antes, gestores tinham que recorrer a servidores próprios, contratavam desenvolvedores de forma externa, o que gerava riscos, como vulnerabilidades de segurança e falta de manutenção. A plataforma oferece alternativas corporativas, conectores oficiais e acesso seguro às bases e às *APIs*, eliminando a necessidade do *Shadow IT* e garantindo que todas as aplicações estejam dentro dos padrões corporativos. Isso transforma o "barato que sai caro" em uma solução escalável, com atualizações periódicas e suporte corporativo adequado.

#### 5.2.1.2 Perspectiva Estratégica da plataforma *low-code* no contexto de TD

A adoção do APS trouxe à tona importantes desafios culturais dentro da organização, especialmente no que diz respeito à integração de diferentes perfis de desenvolvedores e à adaptação às novas práticas de desenvolvimento digital. De acordo com o gerente estratégico, "a introdução de uma tecnologia disruptiva como o *low-code* enfrentou resistências naturais, especialmente entre profissionais mais experientes que estavam habituados a metodologias tradicionais". Essa resistência é comum em processos de TD, conforme apontado por Verina e Titko (2019), e reflete a necessidade de gestão ativa para garantir que os colaboradores estejam alinhados às mudanças organizacionais.

Os desafios iniciais envolveram principalmente a aceitação e a adaptação por parte dos desenvolvedores com diferentes níveis de experiência técnica. O *citizen developer*, por exemplo, destacou que "a acessibilidade da plataforma permite

que pessoas com pouca formação técnica contribuam no desenvolvimento de soluções, mas isso inicialmente gerou tensões com profissionais mais técnicos, que receavam a perda de controle sobre os processos". Essa tensão foi mitigada por meio de *workshops* e treinamentos internos, que não apenas capacitaram novos usuários, mas também promoveram uma cultura de colaboração e aprendizado mútuo.

Outro aspecto destacado é o papel dos *citizen developers* na mudança do paradigma de desenvolvimento. Ao combinar conhecimentos de negócios com habilidades técnicas, esses profissionais não apenas desafiam as estruturas tradicionais, mas também oferecem um meio termo entre as demandas estratégicas e as soluções tecnológicas. Segundo o gerente de desenvolvimento, "os *citizen developers* trouxeram uma perspectiva renovada para o processo de criação de soluções, ao mesmo tempo em que demandaram ajustes nos padrões de qualidade e nas políticas internas de governança". Essa mudança cultural está alinhada com tendências globais de democratização do desenvolvimento tecnológico (SANCHIS et al., 2019).

A colaboração entre áreas técnicas e de negócios foi amplamente facilitada pela plataforma. O APS proporcionou um ambiente no qual diferentes áreas puderam se engajar em projetos conjuntos, quebrando barreiras organizacionais que antes dificultavam a troca de informações. Para o gerente estratégico, "o *low-code* criou um espaço comum onde equipes técnicas e não técnicas podem colaborar de forma mais eficiente, traduzindo requisitos de negócios em soluções digitais com rapidez". Essa integração é um reflexo de uma tendência maior na digitalização das organizações, conforme observado por Naqvi et al. (2023), em que plataformas de desenvolvimento visual são utilizadas como ponto de convergência entre diferentes departamentos.

Contudo, os entrevistados destacaram que nem todos os desafios culturais foram superados. Em algumas áreas mais conservadoras, a aceitação do *low-code* ainda encontra resistências, particularmente entre profissionais que possuem uma visão tradicionalista da tecnologia. Essa resistência foi abordada por meio de iniciativas como laboratórios de inovação, que serviram como ambientes controlados para testar soluções e demonstrar o valor da plataforma de forma tangível. Segundo

o *citizen developer*, "esses laboratórios foram essenciais para reduzir as barreiras culturais e fomentar a confiança gradativa na plataforma".

O APS também se destacou na sua capacidade de incorporar perspectivas diversificadas no desenvolvimento de soluções. Para o gerente de desenvolvimento, "a plataforma permitiu que diferentes vozes fossem ouvidas no processo de criação, promovendo um ciclo de *feedback* mais rápido e eficaz". Essa inclusão de múltiplos perfis ampliou o escopo de inovação dentro da organização, permitindo que soluções fossem criadas de forma mais alinhada às necessidades reais do negócio e às expectativas do mercado (NAQVI et al., 2023).

Finalmente, o sucesso na integração de diferentes perfis foi percebido como um catalisador para a adoção mais ampla do *low-code*. Ao reduzir as barreiras entre equipes técnicas e não técnicas, a plataforma não apenas acelerou a entrega de soluções, mas também promoveu uma mentalidade mais colaborativa e inovadora. Essa mentalidade é essencial para o sucesso de qualquer transformação digital, consolidando o *low-code* como uma abordagem estratégica para a evolução organizacional.

#### 5.2.1.3 TD impulsionada pelo *low-code*

As entrevistas evidenciaram o papel central do *low-code* como catalisador da TD na organização, com impacto direto na mitigação do *Shadow IT* e na promoção de governança eficaz. Tanto o gerente estratégico quanto o *citizen developer* enfatizaram que a centralização das soluções em uma plataforma *low-code* reduziu significativamente as redundâncias tecnológicas e aumentou a transparência dos processos, avançando em direção aos objetivos organizacionais de digitalização e segurança (BEGONHA; KOPPER; THIRAKUL, 2022).

Um dos aspectos mais marcantes foi o fortalecimento da governança proporcionado pela plataforma. De acordo com o gerente estratégico, "ao padronizar soluções em uma única plataforma, conseguimos mitigar riscos associados ao *Shadow IT*, trazendo maior segurança e confiabilidade para as operações". O *citizen developer* corroborou essa visão, apontando que a centralização na nuvem não apenas eliminou processos manuais propensos a erros humanos, mas também melho-

rou a conformidade com políticas internas e regulações externas (ZIMMERMANN; RENTROP, 2014).

Adicionalmente, o impacto do *low-code* na simplificação da integração entre áreas de negócio e TI foi amplamente destacado. Segundo o gerente de desenvolvimento, "a plataforma criou um espaço comum onde equipes técnicas e não técnicas podem colaborar de forma mais eficiente, traduzindo requisitos de negócios em soluções digitais com rapidez". Por um lado, essa integração é essencial para alinhar as soluções às demandas estratégicas e operacionais da organização, promovendo maior agilidade e inovação (CABOT, 2020).

Por outro lado, desafios relacionados às políticas de acesso foram apontados pelos entrevistados. O *citizen developer* destacou que, embora a governança tenha sido aprimorada, as restrições de acesso muitas vezes retardam o desenvolvimento de soluções ágeis. Contudo, o gerente de desenvolvimento argumentou que essas restrições são necessárias para garantir a segurança dos dados e conformidade com normas regulatórias. Essa tensão entre agilidade e segurança é uma constante em processos de transformação digital, exigindo ajustes contínuos para equilibrar essas prioridades (SANCHIS et al., 2019).

Outro aspecto relevante é o impacto do *low-code* na gestão de conformidade e auditoria. A centralização das soluções em uma plataforma única facilita a rastreabilidade e o monitoramento de processos, conforme destacado pelo gerente de desenvolvimento: "A plataforma possibilita auditorias mais eficazes, pois todos os dados e fluxos estão organizados e acessíveis em um só lugar". Essa característica é especialmente valiosa em setores altamente regulamentados, onde a conformidade é um requisito crítico para operações (ZIMMERMANN; RENTROP, 2014).

Além disso, a utilização do *low-code* foi apontada como um fator de aceleração para a implementação de iniciativas estratégicas de TD. O gerente estratégico mencionou que "o tempo de desenvolvimento foi reduzido consideravelmente, permitindo que a organização respondesse rapidamente a mudanças do mercado e demandas regulatórias". Essa agilidade está alinhada com os pilares da TD, que buscam promover flexibilidade organizacional e eficiência operacional (CABOT, 2020).

Os entrevistados também discutiram o papel do *low-code* na capacitação e inclusão de novos perfis na TD. O *citizen developer* destacou que "a plataforma democratizou o desenvolvimento, permitindo que colaboradores com diferentes níveis de conhecimento técnico contribuíssem para a criação de soluções". Essa democratização é fundamental para expandir as capacidades de inovação da organização, integrando perspectivas diversificadas no desenvolvimento de soluções (NAQVI et al., 2023).

Portanto, o uso de *low-code* revelou-se essencial para impulsionar a TD na organização, proporcionando governança reforçada, eficiência operacional e maior integração entre diferentes departamentos. Embora desafios como restrições de acesso e ajustes culturais persistam, os benefícios superam significativamente as dificuldades, consolidando o *low-code* como um pilar fundamental na evolução digital da organização. Esse modelo não apenas promove maior alinhamento estratégico, mas também posiciona a organização de forma competitiva no cenário digital em constante mudança.

#### 5.2.1.4 Aspectos de Desenvolvimento de Software com *low-code*

Com o uso da plataforma para desenvolvimento de software, nota-se *insights* sobre diversos aspectos desde qualidade de software, integração com sistemas externos, adoção de requisitos, até gerenciamento de *backlog*. Nesse sentido, de acordo com o gerente de desenvolvimento, e em conformidade com o ilustrado por Sanchis et al. (2019), o tempo de desenvolvimento de uma aplicação, da concepção até a implantação, é um dos principais pontos notados com o uso de plataformas *low-code*. Para o gerente de desenvolvimento, os fatores que mais influenciam na mudança do *lead time* de uma aplicação, tendo em vista o contexto de TD na *Organização*, reside na crescente habilidade dos desenvolvedores e o acesso a recursos facilitadores providos pela plataforma.

Ainda, para o gerente de nível estratégico, a autonomia do *citizen developer* reduziu a necessidade de aprovações em múltiplos níveis, o que torna suficiente a aprovação do ponto focal do time para que uma aplicação vá ao ar. Além disso, componentes pré-formatados e integrações já configuradas aceleram significativamente o tempo de entrega. O próprio desenvolvedor, por conhecer as regras de



negócio, pode validar e homologar a aplicação diretamente, sem depender de intermediações de outras pessoas. Por conseguinte, o *citizen developer* ratifica esse ponto de vista, ao afirmar que a plataforma permite a centralização de todo o processo operacional para o desenvolvimento da aplicação, tornando desnecessário a alteração entre diferentes sistemas, além de salientar o benefício do uso de recursos pré configurados, o que agiliza a modelagem do aplicativo. Dessa forma, nota-se a forma pela qual as soluções *low-code* aceleram a implementação de soluções digitais (CABOT, 2020).

Nessa linha, o gerente estratégico da plataforma, na *Organização*, destaca a relação do gerenciamento de *backlog*, visando à relação entre entregas com prazos emergenciais e o uso da abordagem *low-code* para o desenvolvimento. Ele destaca que o escopo principal de aplicações *low-code* reside em *back-office* e são direcionadas para automações de tarefas de processos internos, para agilizá-los e entregar maior valor à *Organização*. Também é destacado por ele, como a qualificação eficaz dos desenvolvedores da plataforma ainda é um ponto relevante para a *Organização* na gestão de prazos de desenvolvimento e implantação das aplicações (ROKIS; KIRIKOVA, 2023).

No contexto de desenvolvimento tradicional e do desenvolvimento com a abordagem *low-code* com o uso do APS, foi levantado, junto ao gerente de desenvolvimento e ao gerente de nível estratégico, fatores sobre a integração de aplicações feitas, na plataforma, com aplicações de sistemas externos da *Organização*. Foi observado que essa integração pode gerar gargalos, especialmente na identificação e uso correto de *APIs* - do inglês: *Application Programming Interface*. Além disso, é relevante balancear decisões sobre quais plataformas usar para evitar *lock-in* e garantir que as soluções sejam escaláveis e interoperáveis. Muitas vezes foi necessário desenvolver novos padrões de integração para unir soluções legadas e modernas dentro do ambiente da *Organização*, o que exigiu, de fato, esforços adicionais (VENDRELL-HERRERO et al., 2024).

Nesse contexto, de acordo com o gerente de time, a interoperabilidade da plataforma, muitas vezes, torna-se gargalo em algumas aplicações que demandam *tokens* de autenticação e outros requisitos, de acordo com a política de segurança da *Organização*. Nessa linha, o gerente de nível estratégico, mostrou que o desen-

volvimento de uma nova arquitetura para inteligência artificial generativa está no *backlog* deles. Essa tecnologia será usada para simplificar a descoberta de APIs e facilitará a integração da plataforma com ambientes externos.

A qualidade de software também mostra-se um aspecto relevante para a adequação da plataforma aos padrões da *Organização*. Para o gerente de desenvolvimento, o uso do *Health Assessment* da plataforma permite monitorar métricas relacionadas à qualidade de software, como, por exemplo, segurança, escalabilidade, confiabilidade, entre outros. Nesse cenário, a aferição e medição, realizada com o uso das métricas *DORA*, que avaliam desempenho de desenvolvimento por meio de indicadores como frequência de *deploys*, tempo de *lead* das aplicações, taxa de falhas e tempo de recuperação, podem ser aplicadas à plataforma *low-code*. Assim, mostra-se que a qualidade de software é um aspecto levado em consideração nas aplicações desenvolvidas dentro da plataforma, com o uso do APS na *Organização*.

A equiparação das esteiras de desenvolvimento tradicional é um ponto central para a escalabilidade e utilização da plataforma na *Organização*. Consoante o gerente de desenvolvimento, esse aspecto ainda está sendo amadurecido. Todavia, a possibilidade de empacotar toda a aplicação e, em um único *commit*, ela ser implementada em um novo ambiente, com *logs* e todas as informações necessárias aos desenvolvedores, é um ponto importante para as janelas de oportunidades na *Organização*. Isso mostra, que, de fato, aplicações de *back office* podem ser desenvolvidas e terem o ciclo de vida da aplicação monitorado de maneira mais democrática, ou seja, por *citizen developers*.

Outro aspecto relevante ao desenvolvimento de software na plataforma *low-code* é o uso de recursos de infraestrutura de TI. De acordo com o gerente estratégico, a adoção de nuvem pública reduz custos operacionais, uma vez que o custo de rodar aplicações na nuvem é menor do que mantê-las internamente. Nesse ínterim, para o *citizen developer*, a computação em nuvem também é um fator crucial para o engajamento no uso da plataforma, tendo em vista a unificação da configuração de aplicações em diferentes máquinas, o que alinha o time para o desenvolvimento e promove a *cocriação* de valor (ROKIS; KIRIKOVA, 2023).

Aspectos relacionados ao desenvolvimento de software na plataforma *low-code*, com o uso do APS, na *Organização* são importantes para a análise da TD, sob a perspectiva de *Design de Serviços*. A plataforma destaca-se ao oferecer ferramentas que conciliam qualidade de software, interoperabilidade com sistemas externos e a promoção da interação entre diferentes desenvolvedores, fomentando a criação de valor de forma colaborativa. A possibilidade de equiparar a esteira de desenvolvimento *low-code* com a tradicional, ao mesmo tempo em que se beneficia de recursos pré-configurados e componentes padronizados, demonstra como essa abordagem torna o desenvolvimento mais acessível e inclusivo, sem comprometer a qualidade técnica.

Por fim, a abordagem *low-code* promove uma tecnologia democrática, ao empoderar *citizen developers* e descentralizar processos, garantindo agilidade nas entregas e maior alinhamento com as necessidades da *Organização*. Esse equilíbrio entre simplicidade e eficiência técnica é o fator-chave para engajar pessoas e impulsionar uma TD eficaz, alavancando a inovação e a escalabilidade de soluções de forma sustentável. Assim, de acordo com os pontos focais da *Organização*, a plataforma não apenas acelera a implementação de soluções digitais, mas também reforça o valor estratégico da tecnologia como facilitadora da modernização organizacional.

#### 5.2.1.5 Princípios de Design de Serviços para a promoção da TD

As entrevistas revelaram que o DS desempenha um papel fundamental na promoção da TD dentro da *Organização*, especialmente no contexto da plataforma *low-code*. Os princípios de DS, como empatia, *cocriação* e iteração, emergem como elementos cruciais para garantir que soluções tecnológicas sejam alinhadas às necessidades reais dos usuários e dos objetivos organizacionais. Conforme apontado pelo gerente estratégico, "o sucesso da TD não está apenas na implementação de tecnologias, mas também em como essas tecnologias são projetadas para atender às demandas do negócio e dos colaboradores".

Um dos aspectos mais destacados foi a utilização do APS para fomentar a *cocriação* de soluções entre equipes técnicas e de negócios. O *citizen developer* ressaltou que "a plataforma permite que diferentes perfis de profissionais contri-

buam em projetos compartilhados, utilizando princípios de design centrados no usuário para desenvolver soluções que sejam intuitivas e eficazes". Essa abordagem não apenas promove maior engajamento, mas também aumenta a adesão às ferramentas desenvolvidas, gerando maior valor para a *Organização* (SANDERS; STAPPERS, 2008).

O gerente de desenvolvimento também destacou a importância da iteração e do *feedback* contínuo como princípios do DS. Ele afirmou que "as soluções desenvolvidas na plataforma *low-code* passam por ciclos rápidos de validação, garantindo que os ajustes necessários sejam feitos antes da implantação definitiva". Essa prática é especialmente relevante em um ambiente dinâmico, em que as necessidades dos usuários podem mudar rapidamente, exigindo flexibilidade por parte das equipes de desenvolvimento (PATRICIO; FISK, 2013).

Outro ponto levantado pelos entrevistados foi o papel da empatia como base para o DS na TD. O gerente estratégico observou que "a compreensão profunda das necessidades e desafios enfrentados pelos usuários finais é essencial para projetar soluções que realmente agreguem valor". Essa abordagem foi particularmente importante no desenvolvimento de soluções voltadas para o uso interno da organização, em que o foco está na eficiência operacional e na melhoria da experiência do colaborador.

A colaboração entre departamentos foi outro aspecto em que os princípios de DS se mostraram fundamentais. O APS serviu como um ponto de conexão entre diferentes áreas, permitindo que ideias fossem compartilhadas e refinadas de forma coletiva. Segundo o *citizen developer*, "a capacidade de envolver múltiplos *stakeholders* no processo de desenvolvimento não apenas melhora a qualidade das soluções, mas também garante maior alinhamento com os objetivos estratégicos da *Organização*" (STICKDORN et al., 2022).

Dessa forma, a implementação dos princípios de DS na plataforma *low-code* contribuem significativamente para a promoção da TD ao estabelecer uma cultura de experimentação e aprendizado. Como destacou o gerente de desenvolvimento, "a possibilidade de criar protótipos rapidamente e testar soluções em escala reduzida permitiu que a *Organização* aprendesse com os erros e ajustasse o curso

quando necessário". Essa prática é um elemento-chave para garantir a agilidade e a resiliência da *Organização* em um ambiente de negócios cada vez mais dinâmico (TRISCHLER; PERVAN; SCOTT, 2017).

Em síntese, os princípios de DS se mostraram indispensáveis para o sucesso da TD promovida pelo *low-code*. Ao integrar empatia, *cocriação* e iteração no processo de desenvolvimento, a organização conseguiu alinhar suas soluções tecnológicas aos objetivos estratégicos e às necessidades dos usuários, consolidando o *low-code* como um elemento central em sua evolução digital.

### 5.2.2 Análise Documental

A análise documental desempenha um papel central neste estudo, fornecendo subsídios quantitativos e qualitativos que complementam os dados obtidos nas entrevistas e aprofundam a compreensão sobre o impacto do *low-code* na *Organização*. Feita utilizando relatórios, *dashboards* e outros documentos, esta etapa busca explorar como os processos, práticas e resultados se alinham aos objetivos estratégicos e às metas de TD promovidas pelo APS.

Esta abordagem possibilita a triangulação de dados, permitindo validar os achados qualitativos com evidências objetivas extraídas dos documentos. Além disso, fornece uma perspectiva detalhada dos indicadores-chave de desempenho (KPIs) e de outros elementos operacionais, como tempo de desenvolvimento, adoção da plataforma e eficiência financeira.

Os documentos analisados incluem:

- **Relatórios**, com informações sobre KPIs, como tempo de desenvolvimento, custos e qualidade das soluções entregues;
- ***Dashboards* organizacionais**, que ofereceram métricas em tempo real, como número de aplicações desenvolvidas, taxas de adoção pelos usuários e indicadores de produtividade;
- **Pesquisas internas**, que refletiram a percepção dos colaboradores sobre o impacto do *low-code* e a usabilidade do APS.

Essa análise permite identificar padrões, tendências e correlações entre as práticas de desenvolvimento adotadas e os resultados obtidos, reforçando a base de evidências para a validação das conclusões do estudo.

Especificamente, a análise documental investiga o impacto da implementação do APS no ciclo de desenvolvimento e na entrega de soluções tecnológicas, analisando também a adoção e o engajamento dos colaboradores com as ferramentas disponibilizadas. Além disso, busca compreender a contribuição da plataforma *low-code* para a mitigação de riscos organizacionais e o aumento da produtividade, enquanto explora evidências quantitativas de *savings* e melhorias operacionais, conectando os dados documentais aos objetivos específicos do estudo.

Os critérios de seleção dos documentos foram cuidadosamente estabelecidos para garantir que as informações analisadas fossem relevantes, representativas e confiáveis. Foram considerados:

- **Relevância para os objetivos específicos:** Apenas documentos que demonstrassem ligação direta com os resultados esperados e os propósitos do estudo foram incluídos;
- **Representatividade dos processos analisados:** Garantiu-se que os documentos cobrissem uma ampla variedade de áreas e práticas organizacionais, refletindo diferentes contextos e cenários de aplicação;
- **Atualidade e confiabilidade:** Foram priorizados relatórios recentes, atualizados e provenientes de fontes oficiais, minimizando possíveis distorções ou inconsistências;
- **Integração com outros dados:** Documentos que pudessem ser cruzados com as entrevistas e outras análises qualitativas foram priorizados, permitindo maior riqueza na triangulação dos resultados;

A seguir, a apresentação dos achados e a análise documental são detalhados em subseções específicas.

### 5.2.2.1 Padrões e Tendências Identificados

A análise documental revelou padrões significativos e tendências relacionadas à implementação do APS como ferramenta *low-code*, evidenciando mudanças no ciclo de desenvolvimento, adoção organizacional e impacto financeiro. Esses achados foram complementados pelos dados qualitativos das entrevistas e apoiados pelo referencial teórico, demonstrando o alinhamento estratégico da plataforma com as demandas de TD. Seguem abaixo os tópicos de discussão e descoberta condensados:

- **Eficiência Operacional:** Os *dashboards* analisados indicaram uma redução substancial nos tempos médios de desenvolvimento. Antes da adoção do APS, projetos de médio porte demandavam, em média, 120 dias para serem concluídos; com o uso da plataforma, esse prazo foi reduzido para 75 dias, representando uma diminuição de 37,5%. Na reunião sobre métricas, foi destacado que essa redução reflete diretamente a utilização de componentes pré-configurados e a centralização dos fluxos de trabalho, eliminando redundâncias.

Adicionalmente, a métrica de esforço (SES) apresentou uma média de 4,2, indicando que a maioria dos usuários considera o esforço para executar ações como baixo. Dados detalhados mostram que 48% atribuíram nota 5 (esforço mínimo), enquanto 35% deram nota 4. Esse alto desempenho foi atribuído à acessibilidade das ferramentas e à facilidade de documentação, aspectos mencionados tanto nos *dashboards* quanto nas entrevistas.

Segundo [Paternò \(2013\)](#), a redução de complexidade no desenvolvimento de software é um dos pilares das plataformas *low-code*, o que permite maior foco na entrega de valor ao usuário final. Nas entrevistas, esse ganho de eficiência foi frequentemente mencionado como um dos principais benefícios percebidos pelos desenvolvedores, que destacaram a simplificação de processos antes burocráticos. Adicionalmente, o impacto dessa eficiência se refletiu em prazos mais curtos de homologação e menor dependência de processos manuais.

- **Adoção da Plataforma:** A análise documental e as discussões internas evidenciaram um crescimento constante na adoção do APS, com 85% dos

desenvolvedores utilizando a plataforma em pelo menos um projeto dentro de dois anos de sua implementação. *Workshops* e treinamentos foram citados como estratégias-chave para esse sucesso. Durante a reunião, foi mencionado que a comunidade interna da plataforma atingiu 15.732 membros, com um aumento de 25% em visualizações de conteúdo no último trimestre, totalizando 35.000 visualizações.

Esse engajamento é demonstrado pelos indicadores de Pontuação de Execução da *Sprint* (em inglês, Sprint Execution Score, SES) e de Satisfação de Execução da *Sprint* (em inglês, Sprint Execution Satisfaction, SESAT), que registraram uma média de 4.6. A maioria dos usuários (55%) atribuiu nota 5, enquanto 30% deram nota 4, destacando a alta satisfação com os serviços oferecidos pela plataforma.

Esse movimento de adoção e engajamento está alinhado aos princípios de EUD descritos por [Lieberman, Paternò e Wulf \(2006\)](#), que destacam a importância de ferramentas intuitivas para promover maior autonomia dos usuários. Além disso, os entrevistados citaram que a plataforma permitiu maior colaboração entre times multifuncionais, reforçando a integração entre as áreas de negócio e TI. Essa colaboração foi mencionada como essencial para a identificação de demandas emergentes e para o desenvolvimento de soluções mais aderentes às necessidades operacionais.

- **Impacto Financeiro:** A plataforma APS gerou *savings* concretos, com uma redução média de 35% no tempo de desenvolvimento ao comparar abordagens *low-code* e *no-code* e tradicionais. Esse impacto foi detalhado durante as reuniões, em que foi destacado que a importância da automação de processos e a integração eficiente entre times reduziram retrabalhos e custos operacionais.

Adicionalmente, os dados mostraram uma redução de 15% no índice de falhas em projetos entregues, minimizando custos de manutenção corretiva. Durante as discussões, gestores financeiros reforçaram que esses *savings* permitiram reinvestimentos estratégicos em tecnologias emergentes, fortalecendo a posição da *Organização* como líder em inovação tecnológica.



Os achados confirmam estudos como o de [Sanchis et al. \(2019\)](#), que apontam o *low-code* como um habilitador estratégico para a TD, capaz de mitigar custos operacionais e aumentar a produtividade. Durante as entrevistas, os gestores financeiros ressaltaram que o APS não apenas reduziu custos, mas também contribuiu para justificar investimentos em outras tecnologias emergentes, criando um ciclo virtuoso de inovação organizacional. O alinhamento entre os ganhos financeiros e os indicadores de desempenho reforça a visão de que plataformas *low-code/no-code* podem transformar a gestão de TI em termos de eficiência e custo-benefício.

#### 5.2.2.2 Divergências e Explicações:

Alguns pontos de divergência foram identificados ao longo do estudo. Primeiramente, a dependência de atualizações manuais em certos relatórios resultou em discrepâncias entre os dados documentais e as percepções dos entrevistados. Os indicadores de adoção do APS, por exemplo, variaram significativamente entre os números relatados em *dashboards* e os relatos qualitativos dos usuários. Além disso, os indicadores SES e SESAT, embora úteis para capturar esforços e satisfação, não refletiram completamente os desafios contextuais mencionados em entrevistas específicas. Por fim, diferenças no foco de análise foram observadas: enquanto os *dashboards* enfatizavam métricas técnicas, os colaboradores destacavam aspectos subjetivos, como a melhoria na integração entre as áreas de TI e negócios.

As divergências observadas ao longo do estudo não foram apenas decorrentes de limitações metodológicas, mas também refletiram diferentes perspectivas e interpretações dos resultados por parte das equipes envolvidas. Um exemplo disso foi a forma como os dados de adoção do APS foram analisados: enquanto os *dashboards* indicavam uma taxa de utilização consistente, alguns colaboradores relataram dificuldades específicas em projetos menores, sugerindo que o uso da plataforma ainda variava de acordo com o contexto operacional e individual.

Outro ponto relevante foi a interpretação dos indicadores SES e SESAT. Embora tenham mostrado resultados majoritariamente positivos, como altos níveis de satisfação e de esforço reduzido, alguns entrevistados apontaram que esses

indicadores não capturavam integralmente desafios específicos, como a necessidade de treinamento mais aprofundado em determinados módulos do APS.

Por fim, observou-se uma divergência no impacto percebido do APS em projetos interdepartamentais. Enquanto as áreas técnicas destacaram a eficiência operacional obtida, os gestores das áreas de negócio enfatizaram a importância da integração promovida pela plataforma, mesmo que essa não tenha sido suficientemente capturada nos *dashboards* analisados. Essas diferenças ressaltam a importância de considerar múltiplas perspectivas para interpretar os resultados de forma mais abrangente.

### 5.2.2.3 Discussão Ampliada dos Resultados

A análise dos documentos revelou uma série de insights estratégicos que destacam o impacto da implementação do APS na organização. Primeiramente, os dados demonstraram que o APS influenciou positivamente a tomada de decisões estratégicas, facilitando a priorização de iniciativas digitais e acelerando os processos de transformação digital. Um dos aspectos mais marcantes foi a correlação entre os savings documentados e a percepção de eficiência operacional, conforme reportado pelos stakeholders. Esses savings refletem a capacidade da organização de otimizar seus recursos, reduzindo custos e retrabalhos, ao mesmo tempo em que aumenta a agilidade das entregas.

A Tabela 4 apresenta de forma estruturada os principais impactos e métricas associadas à adoção do APS, fornecendo uma visão quantitativa das melhorias observadas na organização.

Tabela 4 – Impacto do APS na Organização e Métricas Numéricas

<b>Categoria</b>	<b>Impacto Observado</b>	<b>Métricas Numéricas</b>
<b>Tomada de Decisão</b>	Melhoria na priorização de iniciativas digitais e transformação digital. Maior alinhamento estratégico e agilidade nas decisões.	-
<b>Eficiência Operacional</b>	Correlação entre <i>savings</i> documentados e percepção de eficiência. Redução de custos e retrabalho, otimização de recursos.	Redução de 35% no tempo de desenvolvimento; redução de 15% no índice de falhas em projetos
<b>Shadow IT</b>	Mitigação pela centralização das atividades de desenvolvimento. Maior visibilidade e controle sobre os processos.	-
<b>Engajamento Organizacional</b>	Colaboração mais integrada entre áreas técnicas e de negócios. Maior alinhamento e participação multidisciplinar.	85% dos desenvolvedores utilizaram a plataforma em pelo menos um projeto dentro de dois anos da sua implementação.
<b>Aceleração da Transformação Digital</b>	Redução do tempo médio de desenvolvimento e entrega. Fluidez e sinergia entre diferentes áreas organizacionais.	Tempo médio de desenvolvimento reduzido de 120 dias para 75 dias (-37,5%)
<b>Integração de Processos</b>	Maior conectividade entre sistemas e soluções. Aumento da versatilidade e variabilidade no <i>backlog</i> .	-
<b>Cultura Digital</b>	Promoção da experimentação e adoção de novas tecnologias. Incentivo à inovação e adaptação contínua.	25% de aumento de participação na comunidade interna, com mais de 35.000 visualizações
<b>Satisfação dos Usuários</b>	Alta aceitação da plataforma APS. Melhor experiência do usuário final.	Pontuação de satisfação (SE-SAT) média de 4.6 (55% deram nota 5, 30% nota 4)
<b>Impacto Financeiro</b>	Economia na alocação de recursos. Reinvestimento em inovação tecnológica.	<i>Savings</i> financeiros documentados com redução média de 35% no tempo de desenvolvimento

Além disso, o uso da plataforma *low-code* mostrou-se eficaz para mitigar o *Shadow IT*, um dos principais desafios enfrentados pela governança de TI em organizações de grande porte. Essa mitigação ocorre por meio da centralização das atividades de desenvolvimento, permitindo maior visibilidade e controle sobre os processos. Outro destaque foi a promoção do engajamento entre diferentes perfis dentro da organização. A plataforma possibilitou uma colaboração mais integrada

entre áreas técnicas e de negócios, alinhando-se aos pilares fundamentais da transformação digital. Essa integração também permitiu a construção ágil de soluções de TI e a conexão eficiente com sistemas externos, conferindo maior variabilidade no backlog e versatilidade na implementação de novas funcionalidades.

No que diz respeito às implicações para a transformação digital, as evidências reforçam o papel estratégico do *low-code* em múltiplos aspectos. Primeiramente, foi identificado como um acelerador de processos, simplificando fluxos de trabalho e reduzindo significativamente os tempos médios de desenvolvimento e entrega. Além disso, a plataforma demonstrou ser um facilitador essencial na integração de processos e soluções, permitindo maior fluidez e sinergia entre diferentes áreas da organização. Outro ponto relevante é o impacto positivo do *low-code* na promoção de uma cultura digital, que incentiva a experimentação, inovação e adoção de novas tecnologias.

Por fim, a integração de práticas qualitativas e quantitativas na análise de resultados se mostra essencial para alcançar uma visão mais abrangente e fundamentada dos impactos obtidos. Essa combinação oferece subsídios que fortalecem as decisões estratégicas e ampliam a compreensão dos efeitos da transformação digital em diferentes níveis organizacionais.

## 5.3 Validação dos Resultados

A validação dos resultados obtidos foi realizada de forma iterativa e em múltiplos níveis. Inicialmente, a cada etapa de coleta de dados, as informações foram revisadas e discutidas com os colaboradores diretamente envolvidos, como técnicos, desenvolvedores e gestores de equipes. Esse processo permitiu ajustar as interpretações e assegurar que os dados refletissem a realidade operacional da organização.

No encerramento do estudo, uma validação final foi conduzida em conjunto com o gerente de desenvolvimento, que desempenhou um papel central na supervisão das etapas de implementação do APS e no alinhamento estratégico da plataforma com as metas da *Organização*. Durante essa etapa, os principais achados, incluindo as métricas de SES, SESAT, *savings* financeiros e padrões de adoção,

foram apresentados em detalhes, e as validações qualitativas foram cruzadas com os relatórios documentais e *dashboards* analisados.

A validação envolveu ainda discussões específicas sobre:

- **A precisão das métricas de eficiência operacional**, como o tempo médio de desenvolvimento e a frequência de *deploys* bem-sucedidos;
- **A interpretação dos índices de satisfação e esforço** (SES e SESAT), correlacionando-os com os depoimentos dos colaboradores;
- **A viabilidade das práticas recomendadas com base nos savings**, identificados e sua replicabilidade em contextos futuros.

Essa abordagem colaborativa garantiu que os resultados fossem não apenas os mais precisos possíveis, mas também aplicáveis às decisões estratégicas e operacionais da *Organização*.

### 5.3.1 Limitações Documentadas

Algumas limitações foram identificadas ao longo do estudo. Primeiramente, a dependência de atualizações manuais em certos relatórios resultou em discrepâncias entre os dados documentais e as percepções dos entrevistados.

Os indicadores de adoção do APS, por exemplo, variaram significativamente entre os números relatados em *dashboards* e os relatos qualitativos dos usuários. Os relatos sobre a adoção do APS apresentaram possíveis vieses, uma vez que podem ter sido influenciados pela percepção subjetiva dos entrevistados ou pela falta de representatividade completa das informações coletadas. Além disso, os indicadores SES e SESAT, embora úteis para capturar esforços e satisfação, não refletiram completamente os desafios contextuais mencionados em entrevistas específicas.

Por fim, diferenças no foco de análise foram observadas: enquanto os *dashboards* enfatizavam métricas técnicas, os colaboradores destacavam aspectos subjetivos, como a melhoria na integração entre as áreas de TI e negócios, mudanças organizacionais com o uso da tecnologia e maior controladoria sob os processos internos.

### 5.3.2 Restrições Metodológicas

Como um estudo de caso único, esta pesquisa se concentrou exclusivamente na aplicação da plataforma *ServiceNow* (Capítulo 4) em uma única organização. Essa abordagem restringe a generalização dos resultados para outros contextos organizacionais ou plataformas. Além disso, a dependência de dados específicos do APS limitou a amplitude da análise, uma vez que outras ferramentas ou soluções *low-code* não foram diretamente investigadas.

### 5.3.3 Impactos nas Recomendações Finais

Apesar das limitações, a validação contínua com *stakeholders* e a triangulação de dados qualitativos e quantitativos fortaleceram a confiabilidade dos resultados. Esses achados servem como base para futuras implementações em contextos similares, mas recomenda-se a ampliação do escopo metodológico em estudos futuros para abordar possíveis vieses e diversificar as fontes de dados.

## 5.4 Considerações Finais

As análises realizadas ao longo deste Capítulo evidenciam o impacto da adoção da plataforma *low-code*, com o uso do APS, na Organização. Isso demonstra avanços em eficiência operacional, engajamento dos usuários e impacto financeiro. Os dados documentais, combinados com os achados qualitativos das entrevistas, reforçam a capacidade do APS de transformar processos anteriormente burocráticos ou até mesmo não rastreáveis (*Shadow IT*), em fluxos de trabalho mais ágeis e centralizados. Esse alinhamento entre os resultados observados e o referencial teórico (Capítulo 2) destaca a relevância do *low-code* como um habilitador estratégico para a TD, promovendo maior integração entre TI e as áreas de negócio, ao mesmo tempo em que reduz custos operacionais e retrabalhos.

Apesar das limitações metodológicas identificadas, como a dependência de dados específicos e a ausência de comparações com outras plataformas, os achados fornecem *insights* valiosos para aprimorar o uso do *low-code* na Organização e em contextos similares. A validação contínua com *stakeholders* garantiu a apli-

cabilidade dos resultados às decisões estratégicas e operacionais, enquanto as recomendações finais apontam para a necessidade de ampliar a análise em estudos futuros. Assim, este capítulo contribui para o entendimento do impacto das plataformas *low-code* na gestão e no desenvolvimento de TI, consolidando-se como um referencial para práticas mais eficientes e inovadoras.

## 6 Conclusão e Trabalhos Futuros

### 6.1 Principais Conclusões

A partir deste trabalho, observou-se que a TD é um processo das empresas que transcende a mera adoção de tecnologias específicas, como, por exemplo, a computação em nuvem. O estudo destacou que a TD deve ser abordada como um fenômeno centrado nas pessoas, que demanda não apenas ferramentas tecnológicas, mas também uma cultura organizacional robusta, por meio do **engajamento**. Nesse contexto, a combinação do *low-code* com os princípios de DS — como *co-criação*, empatia e centralidade no humano — emerge como catalisadora dessa transformação. Essa integração promove uma cultura de engajamento que não apenas mitiga problemas como o *Shadow IT*, mas também democratiza a tecnologia, aumenta a produtividade e impulsiona a inovação colaborativa.

Por este trabalho, as plataformas como o *App Engine Studio* (APS), da *ServiceNow*, permitem um desenvolvimento mais ágil e inclusivo, alinhando soluções tecnológicas às necessidades dos usuários e aos objetivos organizacionais. Além disso, observou-se que o *low-code* pode ser um fator-chave na estratégia de TD das organizações, especialmente ao ser utilizado de forma estruturada e integrada aos princípios do DS.

Em alinhamento aos objetivos específicos do trabalho, os dados coletados indicam que o *low-code* contribui para:

- Mitigar o *Shadow IT*, promovendo governança e segurança;
- Gerar, estrategicamente, *savings* financeiros ao reduzir custos operacionais;
- Promover uma cultura de TD na *Organização*;
- Inovações colaborativas;
- Acelerar o desenvolvimento de soluções digitais e funcionalidades organizacionais;



- Fortalecer a aplicação de princípios de DS, como a *cocriação* e a empatia, na evolução do ambiente organizacional.

## 6.2 Contribuições para a Prática Organizacional

Os achados deste trabalho oferecem importantes contribuições para a prática organizacional. A integração do *low-code* com princípios de DS possibilita que as organizações criem soluções tecnológicas mais rápidas, alinhadas e sustentáveis. O uso do *App Engine Studio* (APS) é um exemplo prático de como ferramentas *low-code* podem democratizar o desenvolvimento de software, permitindo a participação ativa de diversos *stakeholders*, incluindo usuários finais sem formação técnica.

O **engajamento** gerado pelo uso da plataforma por diferentes perfis de usuários, desde equipes técnicas até colaboradores sem experiência em desenvolvimento, cria um ambiente organizacional coeso e estável. Essa integração promove uma cultura colaborativa, na qual diferentes perspectivas são valorizadas, aumentando a confiança entre os membros da organização e fortalecendo o alinhamento estratégico. Além disso, esse engajamento contribui para a construção de um ecossistema resiliente, no qual as soluções desenvolvidas refletem de forma mais precisa as demandas e expectativas organizacionais.

A mitigação do *Shadow IT* é uma das principais contribuições práticas observadas. Por meio de plataformas *low-code*, é possível substituir soluções informais e desalinhadas por ferramentas estruturadas, que garantem maior segurança, governança e aderência às estratégias organizacionais. Além disso, o trabalho evidenciou o impacto positivo da *cocriação* e da empatia no desenvolvimento de soluções que atendem às reais necessidades dos usuários, promovendo maior aceitação e eficiência.

Ademais, ao democratizar o acesso à tecnologia, o *low-code* possibilita um aumento significativo no *output* de entregas, reduzindo os ciclos de desenvolvimento e otimizando recursos. Essa abordagem também promove uma cultura de inovação organizacional, essencial para as organizações que desejam se manter competitivas em um mercado dinâmico. Os achados deste trabalho oferecem importantes con-

tribuições para a prática organizacional. A integração do *low-code* com princípios de DS possibilita que as organizações criem soluções tecnológicas mais rápidas, alinhadas e sustentáveis. O uso do *App Engine Studio* (APS) é um exemplo prático de como ferramentas *low-code* podem democratizar o desenvolvimento de software, permitindo a participação ativa de diversos *stakeholders*, incluindo usuários finais sem formação técnica.

Por fim, conclui-se que a adoção de plataformas *low-code* com uma abordagem estruturada e orientada por princípios de DS, por meio do **engajamento** gerado, é essencial para transformar desafios em oportunidades e garantir que as organizações não apenas sobrevivam, mas prosperem na era digital.

### 6.3 Limitações e Trabalhos Futuros

Esta pesquisa reafirma a relevância do *low-code* como catalisador da TD, especialmente quando aliado aos princípios de DS. Ao longo do estudo, evidenciou-se que a TD é, antes de tudo, um processo centrado nas pessoas, demandando o engajamento e a colaboração de toda a comunidade organizacional.

Embora o trabalho tenha alcançado seus objetivos, o escopo se limita ao objeto de pesquisa, isso é, um único setor, uma única organização e uma única plataforma *low-code*.

Vislumbra-se, como trabalhos futuros: investigar outras organizações, plataformas e contextos de aplicação; desenvolver estudos complementares explorando como o *low-code* e o DS impactam organizações em setores distintos, além do setor financeiro; avaliar o impacto do *low-code* pelos atores das áreas de negócios, sem formação em tecnologia da informação, entre outros.

## Referências

- BEGONHA, D.; KOPPER, A.; THIRAKUL, T. Low-code/no-code: A way to transform shadow it into a next-gen technology asset. *McKinsey Digital*, August 2022. Disponível em: <<https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/tech-forward/low-code-no-code-a-way-to-transform-shadow-it-into-a-next-gen-technology-asset>>. Citado 3 vezes nas páginas 30, 36 e 85.
- BEHRENS, S.; GYÖRY, A. Shadow systems and it governance: Bridging user needs and organizational policies. Association for Information Systems (AIS), St.Gallen, Switzerland, 2009. Referenciado em Silic e Back (2014). Disponível em: <<https://www.researchgate.net/publication/275340606>>. Citado na página 29.
- BERGVALL-KÅREBORN, B.; WIBERG, M. User driven service design and innovation platforms. In: SPRINGER. *Proceedings of the International Conference on Human-Computer Interaction*. [S.l.], 2013. p. 3–7. Citado na página 28.
- BOCK, A. C.; FRANK, U. Low-code platform. *Business Information Systems Engineering*, Springer, v. 63, n. 6, p. 733–740, 2021. Citado 6 vezes nas páginas 23, 27, 31, 35, 37 e 46.
- BRATINCEVIC, J.; KOPLOWITZ, R. *The Forrester Wave™: Low-Code Development Platforms For Professional Developers, Q2 2021*. 2021. Citado 7 vezes nas páginas 6, 47, 48, 49, 56, 57 e 58.
- BRERETON, P. et al. Using a protocol template for case study planning. In: *Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*. Swindon, GBR: BCS Learning & Development Ltd., 2008. (EASE'08), p. 41–48. Citado 4 vezes nas páginas 7, 18, 60 e 61.
- BUCAIONI, A.; CICHETTI, A.; CICOZZI, F. Modelling in low-code development: a multi-vocal systematic review. *Software and Systems Modeling*, Springer, v. 21, p. 1959–1981, 2022. Disponível em: <<https://doi.org/10.1007/s10270-021-00964-0>>. Citado 3 vezes nas páginas 6, 46 e 72.
- CABOT, J. Positioning of the low-code movement within the field of model-driven engineering. *Journal of Systems and Software*, Elsevier, v. 163, p. 110561, 2020. Citado 6 vezes nas páginas 36, 40, 41, 43, 86 e 88.

CORESYSTEMS. *Difference between Digitization, Digitalization and Digital Transformation*. 2024. Accessed: 2025-02-17. Disponível em: <<https://www.coresystems.net/blog/difference-between-digitization-digitalization-and-digital-transformation>>.

Citado 3 vezes nas páginas 7, 21 e 22.

FARO, B.; ABEDIN, B.; CETINDAMAR, D. Hybrid organizational forms in public sector's digital transformation: a technology enactment approach. *Journal of Enterprise Information Management*, Emerald Publishing Limited, 2021.

Citado na página 20.

FIGUEIREDO, R. et al. Co-design and co-creation in digital public services: A service design approach. In: Workshop de Computação Aplicada em Governo Eletrônico (WCGE). Porto Alegre: Sociedade Brasileira de Computação, Brasil, p. 87–98, 2024. ISSN 2763-8723. Disponível em: <<https://doi.org/10.5753/wcge.2024.2505>>. Citado na página 24.

FIGUEIREDO, R. et al. Stakeholder interaction in the redesign and creation of digital public services: a service design approach. In: *Proceedings of the XXIII Brazilian Symposium on Human Factors in Computing Systems (IHC 2024)*. ACM, 2024. Disponível em: <<https://dl.acm.org/doi/10.1145/3702038.3702107>>. Citado na página 24.

FOX, S. et al. Co-design of a smartphone app for people living with dementia by applying agile, iterative co-design principles: Development and usability study. *JMIR Mhealth Uhealth*, JMIR Publications, v. 10, n. 1, p. e24483, 2022. Citado 2 vezes nas páginas 25 e 46.

GASPAR, A.; MONTEIRO, J. *As práticas de Shadow IT nas empresas: a visão de profissionais contadores*. Dissertação (Mestrado) — Pontifícia Universidade Católica de São Paulo, São Paulo, Brasil, 2018. Disponível em: <<https://tede2.pucsp.br/bitstream/handle/21967/2/Antonio%20Gaspar%20Monteiro%20Junior.pdf>>. Citado 2 vezes nas páginas 29 e 82.

IRIARTE, I. et al. Service design for digital servitization: Facilitating manufacturers' advanced services value proposition design in the context of industry 4.0. *Industrial Marketing Management*, Elsevier, Insert Volume Number, n. Insert Issue Number, p. Insert Page Range, 2023. Disponível em: <[InsertURL](#)>. Citado 3 vezes nas páginas 6, 41 e 42.

JAIN, A.; DAVIS, K.; MATVITSKY, O. Critical capabilities for enterprise low-code application platforms. *Gartner Research Report*, Gartner, United States, 2024. Disponível em: <<https://www.gartner.com/en/documents/4853531>>. Citado 7 vezes nas páginas 50, 51, 52, 53, 54, 55 e 56.

JIA, Y. et al. Digital servitization in digital enterprise: Leveraging digital platform capabilities to unlock data value. *International Journal of Production Economics*, v. 278, 2024. Disponível em: <<https://doi.org/10.1016/j.ijpe.2024.109434>>. Citado na página 22.

KAGAN, R. S. *The New Digital Enterprise: The Power of Low Code and Collaborative Design*. La Jolla, California USA: Birdrock Press, 2021. Citado 10 vezes nas páginas 12, 15, 24, 31, 32, 33, 35, 37, 47 e 57.

KO, A. J. et al. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, ACM, v. 43, n. 3, p. 1–44, 2011. Citado na página 28.

KRYVINSKA, N.; BICKEL, L. Scenario-based analysis of it enterprises servitization as a part of digital transformation of modern economy. *Applied Sciences*, MDPI, v. 10, n. 1076, p. v2, 2020. Citado 2 vezes nas páginas 21 e 23.

LAITINEN, I.; KINDER, T.; STENVALL, J. Co-design and action learning in local public services. *Journal of Adult and Continuing Education*, SAGE Publications, v. 24, n. 1, p. 58–80, 2018. Citado na página 25.

LEE, H.-Y.; GRINEVICH, V.; CHIPULU, M. Dynamic customer experiences in service design. *European Management Journal*, Elsevier, v. 41, p. 521–533, 2023. Citado na página 14.

LIEBERMAN, H.; PATERNÒ, F.; WULF, V. End-user development: An emerging paradigm. In: *End User Development*. [S.l.]: Kluwer Academic Publishers, 2006. p. 1–8. Citado 4 vezes nas páginas 14, 27, 28 e 95.

MACÍAS, J. A. Development of end-user-centered eud software. In: ACM. *Interaccion*. [S.l.], 2012. p. 123–130. Citado 2 vezes nas páginas 14 e 27.

MALMBERG, L. *Building design capability in the public sector: Expanding the horizons of development*. [S.l.]: Linköping University Electronic Press, 2017. Citado na página 41.

MARSICANO, G. et al. Digital transformation of public services in a startup-based environment: Job perceptions, relationships, potentialities and restrictions. University of Brasilia, Brasilia, Brazil, 2024. Citado na página 20.

MATVITSKYY, O.; DAVIS, K.; JAIN, A. Magic quadrant for enterprise low-code application platforms. *Gartner Research Report*, Gartner, United States, 2024. Disponível em: <<https://www.gartner.com/en/documents/5844247>>. Citado 10 vezes nas páginas 6, 48, 49, 50, 51, 52, 53, 54, 56 e 58.

- MELO, C.; FIGUEIREDO, R.; VENSON, E. Descentralizando ti com a adoção do end-user development: dois estudos de caso. *iSys / Revista Brasileira de Sistemas de Informação*, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), v. 11, n. 1, p. 5–29, 2018. Citado na página 27.
- MERGEL, I.; GANAPATI, S.; WHITFORD, A. B. Agile: A new way of governing. *Public Administration Review*, Wiley Online Library, v. 81, n. 1, p. 161–165, 2021. Citado na página 25.
- MINAYA, P. E.; AVELLA, L.; TRESPALACIOS, J. A. Synthesizing three decades of digital servitization: a systematic literature review and conceptual framework proposal. *Service Business*, v. 18, p. 193–222, 2024. Disponível em: <<https://doi.org/10.1007/s11628-024-00559-x>>. Citado na página 21.
- NAQVI, S. A. A. et al. Understanding the socio-technical aspects of low-code adoption for software development. *Applied Sciences*, MDPI, v. 10, n. 1076, p. v2, 2023. Citado 5 vezes nas páginas 31, 40, 84, 85 e 87.
- OECD. *Going Digital in Brazil*. Paris, 2020. Disponível em: <<https://doi.org/10.1787/e9bf7f8a-en>>. Citado na página 20.
- PATERNÒ, F. End user development: Survey of an emerging field for empowering people. *ISRN Software Engineering*, Hindawi, v. 2013, p. 532659, 2013. Citado 5 vezes nas páginas 27, 28, 31, 47 e 94.
- PATRICIO, L.; FISK, R. P. Understanding the cocreation experience: The collaborative ecosystem of firms and customers. *Journal of Service Management*, Emerald Group Publishing Limited, v. 24, n. 3, p. 223–256, 2013. Citado na página 91.
- RAMIREZ, M. *ServiceNow: Business Model, SWOT Analysis, and Competitors 2023*. 2023. Accessed: 2024-05-30. Disponível em: <<https://pitchgrade.com/companies/servicenow>>. Citado na página 68.
- ROKIS, K.; KIRIKOVA, M. Exploring low-code development: A comprehensive literature review. *Complex Systems Informatics and Modeling Quarterly*, RTU Press, v. 20, n. 36, p. 68–86, 2023. Citado 13 vezes nas páginas 7, 15, 30, 31, 32, 33, 34, 39, 40, 44, 45, 88 e 89.
- SANCHIS, R. et al. Low-code as enabler of digital transformation in manufacturing industry. *Applied Sciences*, MDPI, v. 10, n. 12, p. 1–17, 2019. Disponível em: <<https://www.mdpi.com/2076-3417/10/1/12>>. Citado 16 vezes nas páginas 6, 15, 23, 27, 35, 36, 37, 41, 43, 44, 45, 46, 84, 86, 87 e 96.

SANDERS, E. B.-N.; STAPPERS, P. J. Co-creation and the new landscapes of design. *CoDesign*, Taylor & Francis, v. 4, n. 1, p. 5–18, 2008. Citado na página 91.

SCUPOLA, A.; MERGEL, I. Co-production in digital transformation of public administration and public value creation: The case of denmark. *Government Information Quarterly*, Elsevier, 2022. Citado 2 vezes nas páginas 12 e 20.

ServiceNow. *Exploring App Engine Studio*. 2024. Acesso em: 20 maio 2024. Disponível em: <<https://docs.servicenow.com/pt-BR/bundle/washingtondc-application-development/page/build/app-engine-studio/concept/exploring-aes.html>>. Citado 9 vezes nas páginas 6, 15, 69, 70, 72, 73, 74, 75 e 77.

ServiceNow. *Now Platform*. 2024. Acesso em: 30 maio 2024. Disponível em: <<https://docs.servicenow.com/pt-BR/bundle/washingtondc-now-platform/page/administer/general/concept/now-platform-landing.html>>. Citado 3 vezes nas páginas 15, 68 e 69.

STICKDORN, M. et al. *This Is Service Design Doing: Applying Service Design Thinking in the Real World*. 1. ed. [S.l.]: O'Reilly Media, 2022. Citado 2 vezes nas páginas 13 e 91.

STICKDORN, M.; LAWRENCE, A.; HORMESS, M. *Isto é Design de Serviço na Prática*. Heidelberg, Germany: Springer, 2022. Citado na página 24.

STRONG, D. M.; VOLKOFF, O. Toward understanding enterprise systems configurations: A theoretical framework and research agenda. *Journal of Strategic Information Systems*, Elsevier, v. 13, n. 4, p. 291–323, 2004. Referenciado em Silic e Back (2014). Citado na página 30.

TRISCHLER, J.; PERVAN, S. J.; SCOTT, D. R. The value of co-design: The effect of customer involvement in service design teams. *Journal of Service Research*, SAGE Publications, v. 20, n. 3, p. 240–258, 2017. Citado 3 vezes nas páginas 14, 25 e 92.

TRISCHLER, J.; TRISCHLER, J. W. Design for experience - a public service design approach in the age of digitalization. *Public Management Review*, Taylor Francis, v. 24, n. 8, p. 1251–1270, 2022. Disponível em: <<https://doi.org/10.1080/14719037.2021.1899272>>. Citado 3 vezes nas páginas 24, 25 e 41.

VEERASAMY, B. D. A pragmatic step to deploy low-code web apps on apex cloud services for emerging business assistance. *i-manager's Journal on Software*

*Engineering*, March 2022. Department of Information Technology, University of Technology and Applied Sciences, Shinas, Oman. Citado 2 vezes nas páginas 38 e 39.

VENDRELL-HERRERO, F. et al. The order of the factors matters: How digital transformation and servitization integrate more efficiently. *International Journal of Production Economics*, Elsevier, v. 271, p. 109228, 2024. Citado 5 vezes nas páginas 22, 23, 31, 40 e 88.

VENSON, E.; FIGUEIREDO, R. M. da C.; CANEDO, E. D. Leveraging a startup-based approach for digital transformation in the public sector: A case study of brazil's startup gov.br program. *Government Information Quarterly*, Elsevier, Brasília, Brazil, v. 41, n. 1, p. 101943, 2024. Citado na página 20.

VERINA, N.; TITKO, J. Digital transformation: conceptual framework. *Proceedings of the International Scientific Conference*, Academia.edu, 2019. Citado 3 vezes nas páginas 12, 19 e 83.

WILLSON, S. *The Evolution of ServiceNow: A Look Back and Ahead*. 2023. Accessed: 2024-05-30. Disponível em: <<https://www.xtype.io/general/the-evolution-of-servicenow-a-look-back-and-ahead>>. Citado na página 69.

ZIMMERMANN, S.; RENTROP, C. On the emergence of shadow it – a transaction cost-based approach. In: *Proceedings of the European Conference on Information Systems (ECIS)*. Tel Aviv, Israel: [s.n.], 2014. Disponível em: <[https://www.researchgate.net/publication/263055411\\_On\\_the\\_Emergence\\_of\\_Shadow\\_IT\\_-\\_A\\_Transaction\\_Cost-Based\\_Approach](https://www.researchgate.net/publication/263055411_On_the_Emergence_of_Shadow_IT_-_A_Transaction_Cost-Based_Approach)>. Citado 5 vezes nas páginas 29, 30, 36, 82 e 86.



## APÊNDICE A – Carta de Apresentação

# CARTA DE APRESENTAÇÃO

Brasília, 28 de Novembro de 2024

Organização A/c Responsável Entrevistado

A presente pesquisa representa parte dos requisitos do trabalho de conclusão de curso de graduação de João Pedro de Camargo Vaz e Vinícius Assumpção de Araújo, pela FGA-UnB (Faculdade do Gama – Universidade de Brasília).

Trata-se de um trabalho acadêmico cujo objetivo é investigar como a abordagem Low-code, utilizando o ambiente de desenvolvimento App Engine Studio da ServiceNow, empregada no contexto de Design de Serviços, tem apoiado a estratégia de TD da Organização.

Gostaríamos de ressaltar que todas as informações aqui prestadas serão consideradas em comum acordo com o entrevistado. Qualquer menção explícita será feita somente com a liberação do próprio entrevistado.

As entrevistas foram estimadas para serem respondidas em aproximadamente 90 minutos e se destinam a identificar algumas competências e elementos (processos, tecnologias, habilidades, riscos, benefícios e outros) que estiveram e/ou estão presentes no processo de desenvolvimento de soluções com o uso do Low code/No code.

As entrevistas estão organizadas em blocos com perguntas relacionadas a adoção, implantação e gestão do processo de desenvolvimento usando Low code/No code e seus impactos, quantitativamente e qualitativamente onde esses podem ser observados na Organização. A entrevista é caracterizada como semiestruturada, realizada com tom de conversação.

Na conclusão deste trabalho, o entrevistado receberá uma cópia do estudo.

Desde já, agradecemos a colaboração.

Cordialmente,

João Pedro de Camargo Vaz

Vinícius Assumpção de Araújo

Graduandos em Engenharia de Software

Faculdade do Gama – FGA

Universidade de Brasília – UnB

## APÊNDICE B – Roteiro das Entrevistas

## Roteiro de entrevistas

Objetivo: Investigar o quanto a abordagem Low-code, utilizando o ambiente de desenvolvimento App Engine Studio da ServiceNow, empregada no contexto de Design de Serviços, tem apoiado a estratégia de Transformação digital da Organização

(Ótica do gerente de nível estratégico da plataforma na Organização)

Categoria	Questões
Savings com Solução Low-Code - Nível Estratégico	1. Como a esteira de atendimento de tickets na comunidade da Organização impacta na política de gerenciamento de ativos de TI (ITSM) e como contribui no suporte a serviços digitais e inovações colaborativas? (explicar mais o case de sucesso do fazai).
	2. Como o shadow IT impacta o desenvolvimento de software no BB e como a plataforma low-code melhora esse quadro?
	3. Como o uso da plataforma low-code contribuiu para a integração de Processos Internos e de soluções da Organização? (Criação do Fazai...).
	4. Quais os desafios para o compartilhamento do paradigma de desenvolvimento low code para o restante do banco? Quais os pontos que você acha que contribuem para a não adoção?
	5. Qual estratégia para gerenciar e segregar diferentes soluções no backlog entre diferentes perfis de desenvolvedores na plataforma?
	6. Quais mudanças na utilização de recursos ocorreram com o uso do low-code, para tornar a construção de software mais barata?
	7. Como você avalia o impacto financeiro da implementação da plataforma low-code na organização? Quais economias específicas ou melhorias em recursos você percebeu?
	8. De que forma a plataforma low-code tem contribuído para alcançar metas estratégicas de longo prazo, como expansão de mercado ou inovação?
	9. Na sua percepção, o low-code oferece maior previsibilidade para o planejamento de investimentos em tecnologia?
Tempo de Desenvolvimento para Aplicação Chegar em Produção	1. Sob qual perspectiva macro gerencial o tempo para entregar as aplicações mudou após a introdução do low-code? (Alguma etapa da esteira foi otimizada?)
	2. Existe relação entre entregas emergenciais com o low-code? Tem algum case de exemplo? (Ele falou do padrão de aplicações)

	que são feitas em low-code: são de apoio à decisão e automação, mas não críticas).
	3. Que desafios ainda permanecem em relação ao tempo, mesmo com a utilização do low-code?
Dificuldade: Equiparar Esteira Low-Code com a Esteira Tradicional	1. Quais desafios surgiram ao integrar low-code às abordagens tradicionais de desenvolvimento?
	2. Quais as principais métricas de qualidade observadas pelo time que cuida da qualidade das aplicações? (escalabilidade, manutenibilidade (SaaS), testes, segurança, confiabilidade).
	3. Como ocorre a interoperabilidade entre a plataforma low-code com outros sistemas da Organização? Isso é um gargalo?
	4. O que poderia ser ajustado no processo do low-code para integrá-lo melhor com os outros sistemas da Organização?
Comparação Low-Code e Tradicional, com Métricas e Nível de Aderência para Cada Atributo	1. O quanto e como o uso da plataforma low-code permite o cumprimento de padrões de qualidade da organização?
	2. Quais são os pontos fortes e fracos do low-code em comparação ao desenvolvimento tradicional?
	3. Que desafios surgem ao comparar a aderência a requisitos de negócio e técnicos entre as abordagens?
Transformação Digital	1. Como o uso da plataforma tem agido em relação aos 3 pilares da Transformação Digital mencionados no relatório anual de 2023 do BB?
	- Impulsionar a experiência digital e inovação.
	- Expandir o uso de inteligência analítica e dados para melhorar a experiência do cliente.
	- Entregar valor por meio da digitalização.
	2. Como a plataforma contribui para transformar o portfólio do BB e integrar a geração de valor em novos negócios?
	3. Você percebe maior capacidade de inovação com o uso do low-code?
	4. Quais são os principais desafios para manter inovação e flexibilidade digital utilizando low-code?
	5. Na sua visão, qual é o papel dos colaboradores e dos clientes na transformação digital, levando em consideração o contexto de design de serviços (explicar o que é) na plataforma low-code?
	6. Como o low-code contribui para a inovação no ambiente organizacional? (interação entre diferentes times, suporte, etc).

(Ótica do citizen developer)

Uso de Recursos e Tempo de Desenvolvimento	1. Quais mudanças na utilização de recursos ocorreram com o uso do low-code? (nível de desenvolvedor, com a plataforma fornecendo os recursos para o desenvolvimento da aplicação em um único software).
	2. Como o tempo para desenvolver e entregar aplicações mudou após a introdução do low-code?
	3. Quais fatores você acredita que mais influenciam essa mudança no tempo até a produção?
	4. Como o low-code auxilia na gestão de prazos curtos ou necessidades emergenciais?
	5. Que desafios ainda permanecem em relação ao tempo, mesmo com a utilização do low-code?
Métricas de Qualidade e Comparação com Abordagens Tradicionais	1. Houve alguma dificuldade em relação às métricas de qualidade de software com o uso da plataforma low-code? (escalabilidade, manutenibilidade (SaaS), testes, segurança, confiabilidade).
	2. O quanto e como o uso da plataforma low-code permite o cumprimento de padrões de qualidade da organização?
	3. Quais são os pontos fortes e fracos do low-code em comparação ao desenvolvimento tradicional?
	4. Que desafios surgem ao comparar a aderência a requisitos de negócio e técnicos entre as abordagens? (o que tem que mudar nos requisitos por se tratar do desenvolvimento na SN).
Transformação Digital	1. Como o uso de low-code tem acelerado a transformação digital na organização?
	2. Você percebe maior capacidade de inovação com o uso do low-code?
	3. Na sua visão, qual é o papel dos colaboradores e dos clientes na transformação digital, levando em consideração o contexto de design de serviços (explicar o que é) na plataforma low-code?
	4. Em sua percepção, o uso do low-code contribuiu para acelerar as entregas e aumentar a vazão de produtos digitais na organização?
	5. Como o low-code impactou a eficiência operacional e a capacidade da organização de se adaptar a mudanças?
	6. Você acredita que o low-code desempenha um papel estratégico na transformação digital da empresa? Por quê?
Desenvolvimento no App Engine Studio	1. Quais recursos do app engine studio você considera mais úteis para agilizar o desenvolvimento de software?

	2. Há funcionalidades ou limitações do App Engine Studio que você acredita impactarem a qualidade ou velocidade das entregas?
	3. Como o App Engine Studio auxilia na prototipagem ou experimentação de ideias no desenvolvimento de software?
Cultura Organizacional com o Low Code	1. Como a adoção do low-code afetou a dinâmica e os processos internos do time de desenvolvimento?
	2. Você percebeu alguma resistência cultural à introdução do low-code no time? Se sim, como ela foi superada?
	3. A chegada do low-code trouxe novas expectativas ou responsabilidades para os membros do time? (o que mudou com a chegada do low code?)
Interação entre Diferentes Personas no Time	1. Como os diferentes papéis dentro do time (como desenvolvedores, analistas de negócios e citizen developers) interagem durante o desenvolvimento de aplicações na plataforma?
	2. Você percebe uma maior integração entre equipes de TI e áreas de negócio com o uso do low-code?
	3. Como o low-code influencia a comunicação e o alinhamento entre pessoas com diferentes níveis de habilidades técnicas?
Princípios do Design de Serviço	1. Como o processo interno do time mudou com o uso do low code para gerar respostas às demandas novas dos usuários, conforme o andamento do projeto? Sob quais perspectivas houve mudança?
	2. A plataforma oferece uma prototipação de processos facilitada?
Shadow IT	1. Antes da introdução do low-code, você percebeu a existência de Shadow IT na organização? Se sim, como era gerenciado
	2. Você acredita que o low-code ajudou a reduzir o Shadow IT, alinhando os times com a estratégia de governança da organização?

(Ótica de gerente de time de desenvolvimento)

Categoria	Questões
Integração e Uso de Recursos	1. Como o uso da plataforma low-code contribuiu para a integração de Processos Internos e de soluções da Organização? (Processos de Contratação SaaS).
	2. Quais mudanças na utilização de recursos ocorreram com o uso do low-code? (dentro do seu time).
	3. De que forma a plataforma low-code tem contribuído para alcançar metas estratégicas de longo prazo?

Gestão de Tempo e Qualidade de Software	1. Quais fatores você acredita que mais influenciam essa mudança no tempo até a produção?
	2. Como o low-code auxilia na gestão de prazos curtos ou necessidades emergenciais?
	3. Houve alguma dificuldade em relação às métricas de qualidade de software com o uso da plataforma low-code? (escalabilidade, manutenibilidade (SaaS), testes, segurança, confiabilidade).
	4. Como ocorre a Interoperabilidade entre a plataforma low-code com outros sistemas da Organização? Isso é um gargalo?
	5. O que poderia ser ajustado no processo do low-code para integrá-lo melhor com os outros sistemas da Organização?
	6. O quanto e como o uso da plataforma low-code permite o cumprimento de padrões de qualidade da organização?
	7. Que desafios surgem ao comparar a aderência a requisitos de negócio e técnicos entre as abordagens? (o que tem que mudar nos requisitos por se tratar do desenvolvimento na SN).
	8. Como as métricas DORA, que avaliam desempenho de desenvolvimento por meio de indicadores como frequência de deploys, tempo de lead, taxa de falhas e tempo de recuperação, podem ser aplicadas à plataforma low-code em comparação com a esteira tradicional de desenvolvimento? (equiparação de esteira low-code com esteira tradicional no desenvolvimento)
Transformação Digital e Design de Serviços	1. Cultura Digital: Evoluir o ambiente organizacional, promovendo inovação e equidade. Como o uso de low-code tem acelerado a transformação digital na organização?
	2. Você percebe maior capacidade de inovação com o uso do low-code?
	3. Como o design de serviços influencia o desenvolvimento de soluções low-code na organização?
	4. De que maneira o design de serviços tem contribuído para criar aplicações mais centradas no usuário?
	5. Quais melhorias na experiência do usuário final foram alcançadas com o uso do design de serviços e low-code?
	6. Você acredita que o design de serviços amplia o valor entregue pelas soluções low-code?
	7. Quais são os principais desafios para manter inovação e flexibilidade digital utilizando low-code?
	8. Na sua visão, qual é o papel dos colaboradores e dos clientes na transformação digital, levando em consideração o contexto de design de serviços (explicar o que é) na plataforma low-code?



Shadow IT	1. Como vc percebeu que o shadow it foi mitigado com a introdução do low code?
	2. Como vc enxerga que o low code diminui o custo de transacao dentro do time?
	3. De que maneira o low code impõe a governança de ti na organização e mitiga o shadow it?
	4. As adaptações no modelo de mercado da plataforma low code para se encaixar com o modelo de governança do bb foram tranquilas de implementar na plataforma? Esse alinhamento aconteceu de forma harmônica?

## APÊNDICE C – Carta de Agradecimento

## **CARTA DE AGRADECIMENTO**

Brasília, 28 de Novembro de 2024

Organização A/c Responsável Entrevistado

Gostaríamos de agradecer a sua participação na pesquisa, que teve como objetivo investigar como a abordagem Low-code, utilizando o ambiente de desenvolvimento App Engine Studio da ServiceNow, empregada no contexto de Design de Serviços, tem apoiado a estratégia de Transformação digital da Organização.

Mais uma vez desejamos destacar que as informações fornecidas e liberadas para menções explícitas pelos entrevistados são de grande valia.

Caso seja de seu interesse dirimir alguma dúvida, sinta-se à vontade para procurar-nos. Teremos o prazer em retribuir sua gentileza e estaremos abertos a qualquer comentário, crítica ou sugestão por meio dos canais de comunicação utilizados nas entrevistas.

Cordialmente,

João Pedro de Camargo Vaz

Vinícius Assumpção de Araújo

Graduandos em Engenharia de Software

Faculdade do Gama – FGA Universidade de Brasília – UnB