

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Colegiado do Curso de Graduação em Engenharia de Sistemas

Milton Pereira Bravo Neto

**ENGENHARIA DE SISTEMAS PARA SERVIÇOS
DE DESENVOLVIMENTO “LOW CODE”**

Belo Horizonte
2025

Milton Pereira Bravo Neto

**ENGENHARIA DE SISTEMAS PARA SERVIÇOS
DE DESENVOLVIMENTO “LOW CODE”**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Sistemas da Universidade Federal Minas Gerais, como requisito parcial para o grau de bacharel (a) em Engenharia de Sistemas.

Orientador: Prof. Dr. André Costa Batista

Belo Horizonte
2025

Esse trabalho é dedicado à aquela pessoa.

Agradecimentos

Você pode escrever aqui os agradecimentos a pessoas que contribuíram para a realização do trabalho.

“Aqui vai uma bela e inspiradora frase.”

Resumo

Escreva aqui o resumo do seu trabalho.

Palavras-chave: palavra-chave 1; palavra-chave 2; palavra-chave 3.

Abstract

Translate here the abstract of your work.

Keywords: keyword 1; keyword 2; keyword 3.

Lista de Siglas e Símbolos

Siglas

RPA	Robot Process Automation
ES	Engenharia de Sistemas
SoI	System of Interest
SoS	System of Systems

Sumário

1	Introdução	14
1.1	Objetivos Geral e Específicos	16
1.2	Contribuições e Originalidade	17
1.3	Organização do Trabalho	17
2	Revisão Bibliográfica	18
2.1	Ciclo de Vida	18
2.2	Arquitetura do Sistema	20
2.3	Trabalhos relacionados	21
2.3.1	Low Code Development Cycle Investigation	21
2.3.2	Exploring Low-Code Development: A Comprehensive Literature Review	22
3	Aspectos Socioeconômicos e Humanidades	24
4	Metodologia	25
4.1	Instrumentos e Materiais	26
4.2	Arquitetura do Sistema	27
	Referências Bibliográficas	29

Capítulo 1

Introdução

Ao longo dos anos vivenciando a experiência no mercado de trabalho em diferentes empresas, é possível notar a dificuldade em mapear, seguir e otimizar processos nas organizações. Um fator agravante é quando o processo não é simplesmente para definir operações administrativas, mas também para definir um modelo de serviço prestado, como consultorias, treinamentos e criação de aplicativos. Além dos processos para a concepção do produto os materiais ofertados, a prestação do serviço possui processos próprios que afetam no relacionamento com as partes interessadas bem como no resultados final a ser entregue. O serviço em questão pode ser resumido no desenvolvimento de ferramentas ou soluções para estruturação, automação e/ou digitalização de processos dentro de organizações. Outro ponto que vale ser ressaltado, é a necessidade de um vasto conhecimento em arquitetura de soluções, de ferramentas e técnicas de desenvolvimento para viabilização do serviço e sua manutenção.

Seria legal começar mencionar outros exemplos e depois mencionar o serviço que tem a ver com o seu trabalho. E não precisa falar que vai ser detalhado mais adiante. Esse parágrafo pode vir no final do anterior.

Vale notar que o mercado cria a necessidade do serviço, e este surge de forma orgânica e na maioria dos casos sem uma definição adequada. Um ciclo de vida do sistema de interesse, seja ele um produto ou um serviço, é essencial para padronizar as etapas. A documentação dessas etapas pode ser estabelecida de acordo com os padrões da Engenharia de Sistemas (ES). Acho que ainda pode ser mantido um tom mais geral, como por exemplo: “Um ciclo de vida do sistema de interesse, seja ele um produto ou um serviço, é essencial para padronizar as etapas. A documentação dessas etapas pode ser estabelecida de acordo com os padrões da Engenharia de Sistemas (ES).”

Quando um ciclo de vida não é bem definido, deficiências operacionais nas integrações e relacionamentos entre as partes interessadas são comuns de serem observados. Onde mais pode se destacar essa deficiência é na área de gerenciamento de requisitos e rastreabilidade do sistema. Quando um ciclo de vida não possui nenhuma documentação de requisitos ou rastreabilidade, isso pode afetar tanto as estimativas de esforço e tempo para os desenvolvimentos e possíveis mudanças, quanto deixa uma grande incerteza de impacto em futuras manutenções.

Ao resolver um problema ou instabilidade, podem ser gerados outros que só serão notados pelos usuários finais do sistema.

No jeito mais generalista, poderia ser: “Quando um ciclo de vida não é bem definido, deficiências operacionais nas integrações e relacionamentos entre as partes interessadas são comuns de serem observados.” De uma forma mais generalista: “Onde mais pode se destacar essa deficiência é na área de gerenciamento de requisitos e rastreabilidade do sistema. Quando um ciclo de vida não possui nenhuma documentação de requisitos ou rastreabilidade, isso pode afetar tanto as estimativas de esforço e tempo para os desenvolvimentos e possíveis mudanças, quanto deixa uma grande incerteza de impacto em futuras manutenções. Ao resolver um problema ou instabilidade, podem ser gerados outros que só serão notados pelos usuários finais do sistema.”

Alterações e inclusão de requisitos nas fases finais de desenvolvimento ou depois do sistema desenvolvido são muito mais custosas que se estes tivessem sido refinados e definidos no início. E sem a capacidade de analisar a rastreabilidade do sistema, como dito anteriormente, as estimativas têm pouco fundamento e refletem pouco a realidade. Logo, prazos são mal calculados, métricas são extraídas de maneira errônea e o gerenciamento da equipe fica prejudicado.

Manter o ciclo de vida do sistema de interesse atualizado e alinhado à realidade ajuda a reduzir custos tanto na concepção quanto na operação do sistema. Isso ocorre de forma direta, otimizando os recursos utilizados por meio de processos e etapas bem definidas, e de forma indireta, diminuindo o retrabalho e o tempo gasto em atividades desnecessárias.

Tem sido uma tendência em diversas empresas a criação de times, setores ou áreas focadas na digitalização e automação de processos, rotinas ou atividades repetitivas na empresa. As áreas de negócios [que são colocadas como clientes para esse serviço](#) abrangem equipes como marketing, finanças, pagamentos, segurança da informação, tesouraria, infraestrutura de tecnologia, qualidade, recursos humanos e diversos outros times que desempenham atividades de escritório. Essas equipes podem ser alvo da iniciativa de digitalização e são consideradas os principais interessados, atuando como clientes do serviço prestado.

Este trabalho se propõe a estudar e analisar um sistema dentro de uma empresa multinacional, com diversos ramos de atuação e ativos geradores de receita. A partir do mapeamento do ciclo de vida atual, propostas de revisão e melhoria são investigadas e elaboradas para abordar as deficiências operacionais nas integrações e relacionamentos com as partes interessadas. Em seguida será proposta uma ferramenta para garantir a rastreabilidade do sistema e o gerenciamento dos requisitos de forma mais integrada e dinâmica.

Aqui então você pode começar a falar de uma maneira mais direta o que você se propõe a estudar. Então, você pode começar da seguinte forma: “Este trabalho se propõe a estudar e analisar um sistema dentro de uma empresa multinacional, com diversos ramos de atuação e ativos geradores de receita. A partir do mapeamento do ciclo de vida atual, propostas de revisão e melhoria são investigadas e elaboradas para abordar as deficiências operacionais nas

integrações e relacionamentos com as partes interessadas. Em seguida será proposta uma ferramenta para garantir a rastreabilidade do sistema e o gerenciamento dos requisitos de forma mais integrada e dinâmica.”

Essa próxima frase pode vir antes de você começar a falar diretamente do seu trabalho. “Tem sido uma tendência em diversas empresas a criação de times, setores ou áreas focadas na digitalização e automação de processos, rotinas ou atividades repetitivas na empresa.”

Essa parte aqui eu não sei muito bem aonde encaixar. “O surgimento do time de atuação veio dessa tendência, dentro da grande área de “Digitalização Global” da organização foi montada uma equipe com o objetivo de atender todos esses focos de trabalho, com atendimento disponível para todas as áreas de negócio da América do Sul.”

Esse parágrafo pode vir depois de “Tem sido uma tendência em diversas empresas a criação...” e antes do começo da fala direta do seu trabalho.

O ciclo de vida a ser analisado descreve um serviço interno desta empresa a clientes de digitalização de processos. Esse serviço é o desenvolvimento de uma solução tecnológica que resolva algum problema ou automatize algum processo trazido pelo cliente. É feita uma análise do contexto e realizada uma proposta de resolução para o problema, e em seguida se inicia o desenvolvimento caso isso seja decidido. Em alguns casos, é necessário ainda o suporte à solução desenvolvida, dependendo da complexidade e volume de utilização.

O time de desenvolvimento trabalha numa estrutura de fábrica de aplicativos ou fábrica de software, com poucos desenvolvedores experientes e generalistas, para serem bem engajados com todas as possibilidades a serem exploradas. Esse serviço é repetido a cada nova solução desenvolvida para os times clientes, novos ou não. Diversas tecnologias são utilizadas durante o desenvolvimento, sendo definidas de acordo com a necessidade de cada situação. Entretanto, as duas principais tecnologias são as ferramentas *low code*¹ da empresa Microsoft, conhecidas como ferramentas da *Power Platform*, e códigos em Python para execução de RPAs (do inglês “Robot Process Automation”).

1.1 Objetivos Geral e Específicos

Podemos destacar como objetivos gerais do trabalho realizado a implementação de técnicas e processos de Engenharia de Sistemas para otimizar a prestação de serviços de software de curta duração. **Acho que poderia ser escrito da seguinte forma: “O objetivo geral deste trabalho é melhorar a prestação de serviços de software de curta duração por meio da reforma de processos dentro do seu ciclo de vida.”** Já como objetivos específicos podem ser destacados:

- O mapeamento da situação atual do serviço prestado e do produto entregue.
- A identificação de oportunidades de melhoria no presente cenário.

¹Ferramentas low code são plataformas de desenvolvimento que permitem a criação de aplicativos e sistemas com pouca ou nenhuma necessidade de programação manual. Elas utilizam interfaces visuais, como arrastar e soltar componentes, e configurações pré-definidas para simplificar o processo de desenvolvimento.

- Melhora da definição e levantamento de requisitos no processo existente.
- Elaboração de uma ferramenta para o gerenciamento de rastreabilidade e requisitos.

1.2 Contribuições e Originalidade

Tanto os aspectos de digitalização e otimização de processos quanto o uso de tecnologias *low code* são relativamente novos no que diz respeito à padrões definidos de desenvolvimento, arquitetura ou boas práticas.

O uso de técnicas de ES é interessante pois ao contrário do movimento normal no desenvolvimento de soluções de software que seguem métodos ágeis com muitas iterações e entregas de valor, o processo a ser estudado segue um modelo mais próximo do tradicional modelo de cascata. Entretanto, são conduzidos modelos cascata de curtíssima duração onde os prazos máximos de conclusão variam de 3 a 4 meses, com 3 ou 4 projetos sendo desenvolvidos simultaneamente.

Dessa forma, ao longo do trabalho é desenvolvido um "tailoring" dos conhecimentos da ES para esse contexto e situação de desenvolvimento. **Acho que ficaria melhor da seguinte forma: "Dessa forma, os processos técnicos de ES são adaptados para esse contexto e situação de desenvolvimento."**

1.3 Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma: no Capítulo 2 é apresentada a revisão bibliográfica necessária para o entendimento do trabalho; no Capítulo 3 são apresentados os aspectos socioeconômicos; no Capítulo 4 é apresentada a metodologia utilizada para a realização do trabalho. **Mais pra frente aqui você vai mencionar os capítulos de resultado e conclusão.**

Capítulo 2

Revisão Bibliográfica

2.1 Ciclo de Vida

A definição e criação de um ciclo de vida é uma das formas da Engenharia de Sistemas (ES) atuar no seu propósito de viabilizar o sucesso de um sistema ao mesmo tempo que otimiza a competição existente entre os objetivos das partes interessadas. Ao desmembrar o esforço total e definir os estágios, seus papéis, as novas características do sistema, os critérios de conclusão, os riscos envolvidos e, finalmente, tomar uma decisão, está-se criando o ciclo de vida. Esse processo organiza e estrutura as etapas necessárias para o desenvolvimento e a evolução do sistema de forma clara e eficiente.

Entre cada estágio definido, existem os chamados *decision gates* (portões de decisão, em tradução livre). Nesses pontos, é realizada uma análise do progresso e, como o nome sugere, uma decisão é tomada em relação ao desenvolvimento do sistema.

O ciclo de vida de um sistema é definido a partir de suas características e particularidades, de modo que seus estágios sejam inseridos para atender todas as suas necessidades. Os estágios podem aparecer mais de uma vez, serem executados sequencialmente ou paralelamente e serem inseridos a qualquer momento do ciclo de vida.

Em alguns casos, o Sistema de Interesse (SoI, do inglês *System of Interest*) faz parte de um Sistema de Sistemas (SoS, do inglês *System of Systems*). Nesse contexto, cada um possui seu próprio ciclo de vida. Geralmente, em um SoS, cada elemento do sistema tem seu ciclo de vida independente, e o ciclo de vida do SoS influencia diretamente o do SoI. Por isso, ao analisar o ciclo de vida do SoI, é essencial considerar a evolução e as interações com o SoS.

O ciclo de vida genérico trazido no INCOSE (2023) nos mostra os seis estágios básicos existentes numa estruturação em “V” que busca mostrar de forma visual a aparição desses estágios ao longo do tempo, realçando também o possível paralelismo entre eles. Os estágios são de: conceito, desenvolvimento, produção, utilização, suporte e descontinuação. Na figura ?? podemos ver uma representação do que foi mostrado no livro. **Você pode pegar a imagem do V lá o SEBoK se não der certo com o Handbook.** A seguir cada estágio será abordado:

- **Estágio de conceito:** Este estágio representa a fase exploratória, onde são identificadas as origens de uma necessidade, uma nova missão, uma nova capacidade de negócio ou a alteração de algum desses elementos. Nele, são analisados diversos fatores do sistema, como mercado, aspectos ambientais, condições econômicas, recursos disponíveis e escopo de atuação. O objetivo é definir os limites do problema a ser resolvido, as missões do sistema, onde ele será aplicado e realizar uma análise do negócio, da missão e dos valores que serão entregues. Para garantir uma definição clara do problema, são realizados levantamentos dos requisitos do sistema, das partes interessadas envolvidas e de suas necessidades, além da exploração do espaço de soluções possíveis. Com base nisso, é possível estimar um custo inicial do esforço necessário e criar uma agenda preliminar, que servirá como base para o ciclo de vida do sistema. Alguns dos resultados típicos desse estágio incluem documentos preliminares da arquitetura do sistema, análise de viabilidade, requisitos, design, agenda e estimativa de esforço. Esse estágio é crucial, pois é nele que o sistema é definido. Embora mudanças possam ocorrer posteriormente, sua implementação tende a ser mais complexa e custosa devido a fatores como tempo e recursos adicionais necessários.
- **Estágio de desenvolvimento:** nesse estágio é definido um SoI que atende e vai de encontro com as necessidades e requisitos das partes interessadas, e que pode ser produzido, utilizado, suportado e descontinuado caso necessário. O objetivo principal dessa fase é definir um projeto base de engenharia que pode ser executado, sem buscar a perfeição, mas atendendo às partes interessadas e respeitando os possíveis “trade-offs” previamente definidos nesse mesmo estágio. Nesse projeto base devem estar os requisitos, arquitetura, modelagens, documentação e planejamento para próximas fases que também podem ser vistos como saídas dessa fase.
- **Estágio de produção:** nesse estágio o projeto base definido no estágio anterior sai do papel e dá lugar ao sistema de fato, que será testado e qualificado para ser colocado para utilização.
- **Estágio de utilização:** o início desse estágio se dá com a liberação do sistema ou parte dele para uso, incluindo os sistemas de apoio que são necessários para certas funcionalidades. Esse estágio comumente é o mais longo do ciclo de vida e é comum que mudanças e melhorias no SoI ocorram ao longo da utilização, lembrando sempre de fazer o gerenciamento dos riscos e documentação para garantir a integridade e manutenção do SoI.
- **Estágio de suporte:** segue paralelo ao estágio de utilização assim que alguma funcionalidade se torna disponível, no entanto o preparo e planejamento desse estágio pode ser iniciado antes como a aquisição de sobressalentes. Nesse estágio que são percebidas as melhorias e mudanças que podem vir a ser implementadas durante a utilização.
- Este estágio ocorre quando o sistema é retirado de operação, marcando o fim dos estágios de utilização e produção, ou, no máximo, havendo uma pequena sobreposição entre eles. Além de definir como será realizado o descarte físico ou virtual dos componentes do

sistema, essa etapa também envolve a análise da possível extensão da vida útil de algumas partes do sistema e o arquivamento de documentos importantes relacionados a ele. É uma fase crucial para garantir o encerramento adequado do ciclo de vida do sistema.

Ainda no INCOSE (2023), são trazidos conceitos importante sobre os *decision gates* que coexistem entre os estágio dos ciclo de vida, tanto no início quanto no fim de cada estágio. Dentre os objetivos dos *decision gates* estão o acompanhamento da evolução da maturidade do sistema, a conferência dos critérios de saída ou entrada de um estágio, a análise de risco mediante à situação atual do sistema, e por fim uma tomada de decisão sobre o que será feito. Podendo haver um regresso no ciclo de vida, um avanço, uma pausa ou até mesmo o cancelamento do projeto.

É importante saber equilibrar a formalidade e frequência desses eventos, visto que eles envolvem diferentes partes interessadas, gestores e especialistas, e além disso as decisões devem ser guiadas por dados tomados nos estágios do ciclo de vida e nos artefatos que são gerados para esse momento. Isso evita considerações desnecessárias e inadequadas que podem prejudicar futuramente.

As três abordagens principais trazidas pelo livro para os ciclos de vida são, sequencial, incremental e evolucionário. As principais características dessas três abordagens podem ser resumidas na tabela 2.1 apresentada.

Tabela 2.1: Características das abordagens de um ciclo de vida

Abordadem	Requisitos definidos no início	Iterações planejadas	Múltiplas instalações
Sequencial	Todos os requisitos	Apenas uma	Não
Incremental	Todos os requisitos	Múltiplas	Potencialmente
Evolucionário	Parte dos requisitos	Múltiplas	Tipicamente

Durante a execução dos estágios do ciclo de vida várias tarefas são executadas, e para isso alguns processos precisam ser realizados para garantir a consistência das atividades. Um conjunto de processos é definido no livro e a execução de cada um deles varia de acordo com os estágios existentes no ciclo de vida do sistema.

2.2 Arquitetura do Sistema

Como mencionado na seção 2.1 existem diferentes processos durante o ciclo de vida de um sistema ou projeto. Um deles é o *Processo de Definição da Arquitetura do Sistema*. A existência de um *Estilos de Arquitetura* é de extrema importância para que esse processo seja executado com êxito. Ele atua como um modelo, ou guia, para se construir a arquitetura do sistema. Os *Estilos de Arquitetura* podem ser definidos com base nos ponto de vista da arquitetura, no elementos do sistema e seus relacionamentos, nas conexões, interfaces, mecanismos de interação e possíveis restrições.

Além dos *Estilos de Arquitetura*, outro conceito importante é o de *Padrões de Arquitetura*. Eles são modelos simplificados mas completos no que diz respeito aos elementos do sistema e são reutilizáveis para diferentes tipos cenários. O uso de *Padrões de Arquitetura* agiliza a documentação, facilita a comunicação, promove o reúso, melhora a produtividade e eficiência e serve como um ponto de início para o desenvolvimento de novos sistemas.

Como o conceito de arquitetura pode ser muito abrangente, o SEBoK (SEBoK Editorial Board, 2024) mostra três segmentações de arquitetura, a arquitetura funcional, lógica e a física.

A arquitetura funcional compreende as funcionalidades do sistema, ou seja, quais funções ou comportamentos aquele sistema executa ou possui em diferentes contextos para atingir os objetivos esperados.

2.3 Trabalhos relacionados

2.3.1 Low Code Development Cycle Investigation

O uso de *low code* nas organizações tem diferentes motivos, um deles é o empoderamento de seus funcionários como citado por Pańkowska (2024). Ao qualificar e incentivar os funcionários a utilizarem essas tecnologias para resolverem seus problema, são criadas comunidades de *citizen developers* que em tradução livre pode ser chamado de “desenvolvedores cidadãos”, ou seja, usuários finais que mesmo sem conhecimentos técnicos aprofundados em desenvolvimento de softwares, conseguem criar automações e aplicativos simples para si próprios, devido a baixa barreira de entrada para se adotar essas tecnologias. Dessa maneira, são economizados ou evitados investimentos em infraestrutura e recursos humanos para se manter e criar soluções que utilizam código profissional ou *pro code*, que é muito impactante no orçamento de pequenas e médias empresas.

Ainda em Pańkowska (2024) é introduzido o papel dos “facilitadores *low code*”, que são justamente desenvolvedores profissionais que suportam, capacitam e ajudam a fortalecer a comunidade de *citizen developers*. A autora ainda faz um segregação em dois cenários de desenvolvimento, um para pequenas e médias empresas onde o desenvolvimento é feito inteiramente pelo *citizen developer* e todos os elementos e requisitos do sistema são sua responsabilidade. O outro cenário é mais comum em grandes empresas onde há a integração com sistemas externos ou legados, e assim é necessário o envolvimento das partes interessadas na elucidação dos requisitos e na definição das lógicas de negócio. Esse segundo cenário é mais adequado ao contexto desse trabalho, porém as sugestões apresentadas pela autora são focadas num ciclo de vida gerenciado pelo próprio *citizen developer*, onde ele próprio já sabe os requisitos daquilo que quer desenvolver, e em poucos casos precisa de apoio para definí-los.

Como reforçado, em Pańkowska (2024), o desenvolvimento *low code* não é amplamente utilizado por engenheiros de software profissionais, e há ainda uma falta de padrões e materiais de apoio para a comunidade de desenvolvedores. Para os *citizen developers* esses padrões re-

almente são mais difíceis de serem definidos devido à heterogeneidade dos motivos que levam ao uso do *low code* e mesmo dos próprios desenvolvedores que tem bases de conhecimentos e visões muito diferentes entre si. Todavia, quando se fala de desenvolvimento profissional e a prestação de um serviço, isso se torna mais fácil. Mesmo que o ciclo de vida e processos técnicos definidos não tenham uma sobreposição exata em outros contextos, podem ser modificados ou complementados por outros profissionais antes de serem aplicados.

2.3.2 Exploring Low-Code Development: A Comprehensive Literature Review

O trabalho realizado em Rokis & Kirikova (2023), trás uma síntese do desenvolvimento utilizando ferramentas *low-code* abordando sua definição, suas características, seus benefícios e seus desafios. O autor trás seguinte definição para o desenvolvimento de software *low-code*:

“é uma abordagem de desenvolvimento que melhora o desenvolvimento rápido, flexível e iterativo tornando possível uma rápida tradução dos requisitos de negócios através de uma programação visual com uma interface gráfica, abstração visual, e minimizando a programação manual; envolvendo praticantes com variadas bases de conhecimento e níveis de experiência em desenvolvimento de software.”

Uma plataforma de desenvolvimento *low-code* possui algumas funcionalidades chave que são necessárias para viabilizar um ambiente gerenciável e um desenvolvimento rápido e robusto. Uma dessas funcionalidades indicadas é o suporte à modelagem dos requisitos. A importância dessa funcionalidade segundo o autor é garantir a correta implementação dos requisitos bem como garantem a rastreabilidade e verificação. Vale citar que nem toda plataforma possui todas as funcionalidades, elas na verdade são as mais comuns dentre várias plataformas que serviram de pesquisa.

O ciclo de vida de desenvolvimento *low-code* proposto em Rokis & Kirikova (2023) é bem semelhante a um ciclo de vida ágil, com as seguintes etapas: idealização e análise de requisitos, planejamento, design da aplicação, desenvolvimento, testes, implantação, e manutenção.

Dentre os benefícios citados pelo trabalho para a utilização do *low-code*, está a redução de custo, aceleração do ciclo de desenvolvimento, aumento da responsabilidades ao negócio e mercado, promoção da inovação digital e maior colaboração entre o time de desenvolvimento e negócios.

Em contraste, os desafios para essa abordagem apresentados na etapa de análise de requisitos são a especificação dos requisitos e a constante mudança dos requisitos. E um dos 13 princípios apresentados para o desenvolvimento *low-code* é justamente suportar essa mudança nos requisitos. Isso retém os clientes e vai de encontro com as necessidades do negócio, onde as organizações precisam dessa habilidade para responder a mudanças de mercado e processos. Os autores deixam claro ainda que os desenvolvedores *low-code* devem manter um mente aberta para essas mudanças, permitindo deixar fluir a dinâmica do mercado e do negócio. Pelo fato

dessa abordagem de desenvolvimento ser rápida e ter um ciclo de vida mais curto, isso se torna possível mas continua desafiador.

Capítulo 3

Aspectos Socioeconômicos e Humanidades

A digitalização de processos empresariais é uma estratégia indispensável para aumentar a competitividade, eficiência e segurança nas operações, conforme enfatizado por Sebrae (2023). Estudos apontam que sua adoção pode:

- Elevar a produtividade: Com ganhos de até 30%, ao reduzir em até 90% o tempo para tarefas repetitivas, permitindo que colaboradores se concentrem em atividades estratégicas.
- Aumentar a eficiência: A automação otimiza fluxos de trabalho, com melhorias de até 80% na performance operacional, reduzindo prazos de entrega e aumentando a satisfação dos clientes.
- Reduzir custos: Economias podem chegar a 90% no processamento de dados, 30% na manutenção de equipamentos e 40% na gestão documental.
- Melhorar a experiência do cliente: Processos digitais permitem respostas mais rápidas e personalizadas, podendo aumentar a receita em até 10% e impulsionar o crescimento das empresas em 2,2 vezes.
- Aumentar a segurança da informação: Com controles mais rigorosos, a proteção de dados pode ser ampliada em até 50%, reduzindo riscos e fortalecendo a confiabilidade.
- A digitalização, portanto, transforma não apenas a operação interna das empresas, mas também sua interação com clientes e o mercado, assegurando competitividade e sustentabilidade a longo prazo.

Capítulo 4

Metodologia

Para o TCC2, eu preciso que você reescreva o capítulo mudando o tempo verbal do futuro para o presente ou passado. Ou seja, ao invés de dizer que irá fazer algo, preciso que você escreva que está propondo tal coisa e que tal coisa foi feita da forma X.

Como fundamento do desenvolvimento desse trabalho, está a utilização das técnicas de ES para tornar o dia-a-dia de trabalho e gestão de atividades mais claros e concisos. Além disso, serão especificadas deficiências no processo ou ciclo de vida que, apesar de já serem conhecidas, não estão bem colocadas ou esclarecidas.

Pode ser definido como primeira etapa de trabalho a definição do ciclo de vida como é hoje. Através do estudo do fluxo atual, são definidos os eventos de cada estágio, os entregáveis de cada estágio e os pontos de decisão entre os estágios.

Resumindo, o serviço prestado é basicamente a concepção de diferentes sistemas para atender requisitos específicos em cada caso trazido ao nosso (geralmente, não é colocado em primeira pessoa; você pode escrever como se fosse um observador de fora, mesmo sendo parte da coisa) time, para desenvolvimento e/ou sustentação.

Para ajudar a construir esse ciclo de vida será desenvolvida a arquitetura dos elementos do sistema e estabelecida a relação com as funcionalidades. Assim, pode ser definido todas as opções de possíveis sistemas do serviço prestado, através das combinações de elementos do sistema e das funcionalidades.

Após essa primeira parte do trabalho, com os artefatos e documentações já produzidas, será feita a análise e listagem, dos problemas e deficiências encontradas agora. Focado na parte de gestão de requisitos e rastreabilidade dos sistemas desenvolvidos, serão dados mais detalhes e especificações dos problemas identificados bem como apresentada uma proposta de solução.

Sobre a proposta mencionada anteriormente, se trata do desenvolvimento de uma aplicação *low code* utilizando as ferramentas e recursos da *Power Platform* para correlacionar os elementos do sistema e os requisitos levantados.

Depois de desenvolvida a aplicação, ela será colocada em operação para a coleta de dados de utilização, bem como a percepção dos outros integrantes do time sobre pontos de melhoria ou críticas sobre ela. Métricas de estimativa de esforço de novas funcionalidades ou

alterações podem ser coletadas com mais precisão após essa implementação, pois as relações entre os componentes estão definidas com clareza. Esse resultado poderá ser coletado com uma comparação entre as estimativas para uma mesma tarefa utilizando ou não a ferramenta desenvolvida.

Por fim, será concatenados todos os resultados colhidos para uma análise e avaliação do trabalho realizado. Pontos de melhoria notados no decorrer das atividades realizadas, mas fora do escopo definido, serão indicados para futuras evoluções ou prosseguimento do trabalho.

4.1 Instrumentos e Materiais

A documentação do ciclo de vida atual do sistema de serviço será feita a partir do desenvolvimento de diagrams que representarão os estágios e etapas do ciclo de vida do serviço prestado. Nelas serão destacadas as saídas e entradas. O diagrama será confeccionado na ferramenta online *Draw.io*, que é uma ferramenta gratuita para o desenho de diagramas de diferentes tipos e não restringe o salvamento do arquivo final, mantendo assim a alta qualidade dos diagramas com imagens vetorizadas.

A arquitetura completa das possíveis funcionalidades das soluções desenvolvidas será documentada a partir de blocos de funcionalidades através *Draw.io*. Onde couber, serão estabelecidos os relacionamentos entre as funcionalidades.

A criação da arquitetura geral das soluções desenvolvidas será realizada a partir do mapeamento de todos os possíveis elementos do sistema, sendo esses atômicos ou subsistemas, cobrindo todos os recursos disponíveis para o desenvolvimento das soluções. Também será feito o relacionamento entre esses elementos e suas designações dentre as funcionalidades mapeadas anteriormente. Novamente, o desenvolvimento será também na ferramenta online *Draw.io*.

O levantamento e especificação dos problemas do ciclo de vida atual não requer a utilização de uma ferramenta específica. Ele consiste na análise das entregas das tarefas anteriores, bem como da experiência vivida na rotina em estudo para a definição desses problemas, e então seus detalhamentos. Sendo que a deficiência da gestão de requisitos e da rastreabilidade, já levantada previamente, terá um desenvolvimento mais aprofundado nos conceitos teóricos e referências.

A criação da arquitetura e projeto conceitual do aplicativo será feita a partir da definição de todas as funcionalidades e definições do aplicativo, bem como o registro de sua arquitetura e modelagem dos bancos de dados. Para o registro da arquitetura e da estrutura do banco de dados será utilizado mais uma vez a ferramenta online *Draw.io*.

A aplicação será desenvolvida com as ferramentas e recursos disponíveis na *Power Platform*. Para a interface de usuário será utilizado o *Power Apps*, para automações assíncronas e integrações com sistemas externos será utilizado o *Power Automate*, para a criação do banco de dados será utilizado o *Dataaverse*, que suporta bancos relacionais, e para envios de notificações é utilizado o *Outlook*.

4.2 Arquitetura do Sistema

Na figura 4.1 pode ser observada um padrão de arquitetura física para os sistemas desenvolvidos. Ela contém todos os possíveis elementos do sistema que podem ser utilizados para a arquitetura final de cada projeto executado.

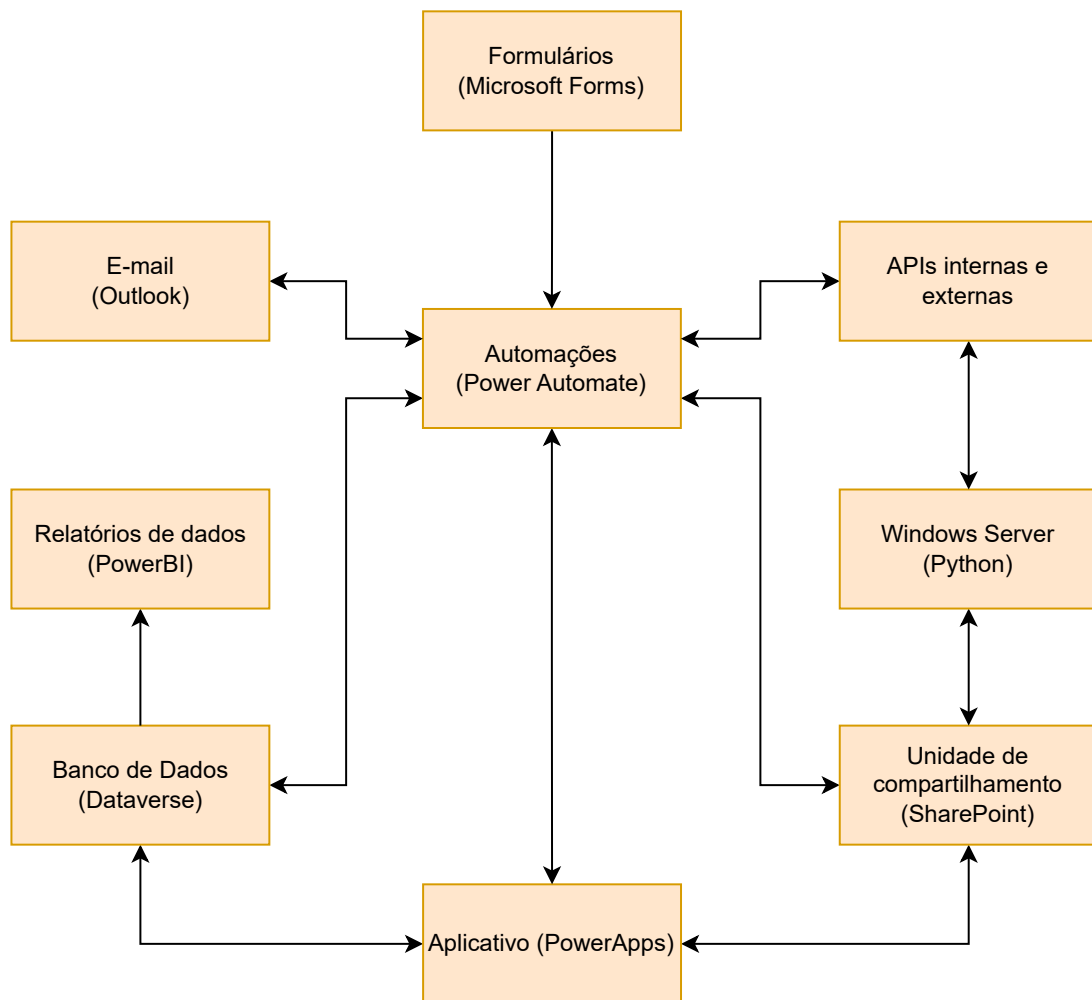


Figura 4.1: Padrão de arquitetura física dos sistemas desenvolvidos.

Já na figura 4.2 temos o padrão de arquitetura funcional para os sistemas desenvolvidos, e mais uma vez, contém todas as possibilidades de funções disponíveis e que podem ser implementadas no sistema de interesse.

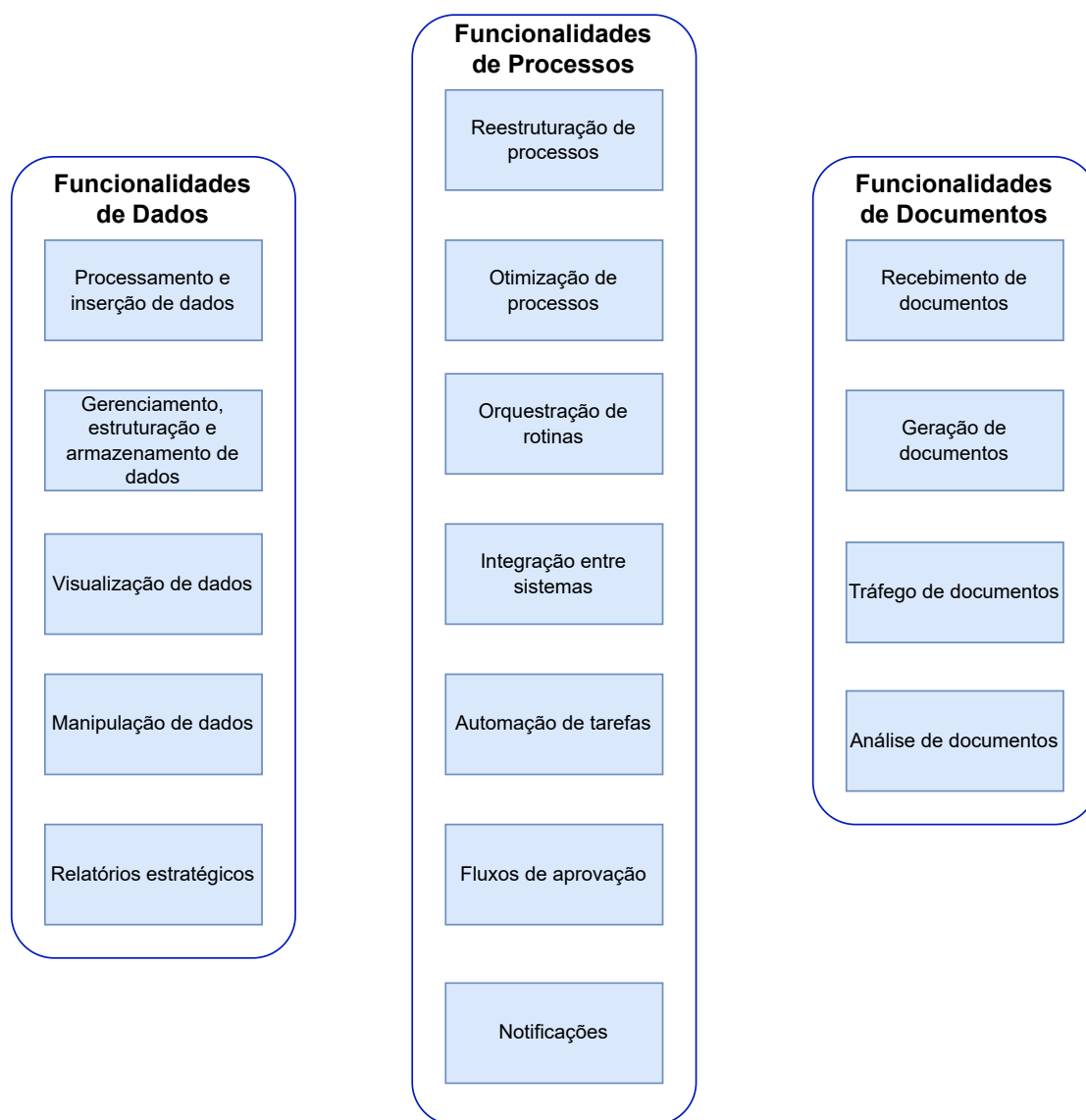


Figura 4.2: Padrão de arquitetura funcional dos sistemas desenvolvidos.

Referências Bibliográficas

- INCOSE (2023). INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. International Council on Systems Engineering (INCOSE), San Diego, CA, 5th edition.
- Pańkowska, M. (2024). Low code development cycle investigation. In Yang, X.-S., Sherratt, S., Dey, N. & Joshi, A., editors, Proceedings of Ninth International Congress on Information and Communication Technology, pages 265–275, Singapore. Springer Nature Singapore.
- Rokis, K. & Kirikova, M. (2023). Exploring low-code development: A comprehensive literature review. page 68–86.
- SEBoK Editorial Board (2024). The Guide to the Systems Engineering Body of Knowledge (SEBoK). The Trustees of the Stevens Institute of Technology, 2.11 edition. Acesso em: 14 jan. 2025. www.sebokwiki.org.
- Sebrae (2023). Como a digitalização de processos impacta os resultados da empresa. Acesso em: 14 jan. 2025.