Advances in Global Innovation & Technology

Recebido
Received
Recibido
Dezembro, 2024
December, 2024
Diciembre, 2024

Aceito
Accepted
Aceptado
Maio, 2025
May, 2025
Mayo, 2025

Publicado
Published
Publicado
Junho, 2025
June, 2025
Junio, 2025

https://**git.fateczl.edu.br**

e_ISSN
2965-3339

DOI
10.29327/2384439.3.3-6

São Paulo
v. 3 | n. 3
v. 3 | i. 3

e33352

Abril-Junho
April-June
Abril-Junio
2025

Arquitetura Workflow para Otimizar Processos em Clínica Oncológica

Workflow Architecture to Optimize Processes in an Oncology Clinic
Arquitectura de Workflow para Optimizar Procesos en una Clínica Oncológica

Igor Amaral Martins[1]
igor.martins5@fatec.sp.gov.br

Wilson Vendramel[1]
Wilson.vendramel@fatec.sp.gov.br

1 – Faculdade de Tecnologia da Zona Leste – Fatec Zona Leste

Resumo:

A gestão eficiente dos fluxos de trabalho é essencial para clínicas de oncologia, visando garantir a qualidade do atendimento e a eficácia dos tratamentos. Este trabalho propõe uma solução baseada em arquitetura workflow para otimizar processos clínicos em uma clínica de oncologia, com foco na eficiência operacional. A proposta é fundamentada em pesquisas documentais e revisões bibliográficas de requisitos clínicos e na definição de uma arquitetura adaptada, utilizando técnicas de modelagem para garantir escalabilidade, segurança e manutenibilidade. Diagramas com notações UML e BPMN são apresentados como base para projetos futuros, visando a unificação e automação dos fluxos de trabalho. Os resultados obtidos cobrem as duas primeiras etapas do ciclo de vida arquitetural (análise de requisitos e design). Embora esta proposta não envolva a implementação completa, ela fornece uma estrutura robusta para o desenvolvimento futuro de sistemas de software funcionais, coesos e manuteníveis, com a necessidade de validações práticas.

Palavras-chave: Workflow; BPMN; Visões Arquiteturais de Software; FURPS+.

Abstract:

Efficient workflow management is essential for oncology clinics, aiming to ensure quality care and treatment effectiveness. This paper proposes a solution based on a workflow architecture to optimize clinical processes in an oncology clinic, with a focus on operational efficiency. The proposal is based on documentary research and literature reviews of clinical requirements, as well as the definition of an adapted architecture, using modeling techniques to ensure scalability, security, and maintainability. Diagrams with UML and BPMN notations are presented as a foundation for future projects, aiming at the unification and automation of workflows. The results obtained cover the first two stages of the architectural lifecycle (requirements analysis and design). Although this proposal does not involve full implementation, it provides a robust framework for the future development of functional, cohesive, and maintainable software systems, with the need for practical validations.

Keywords: *Workflow; BPMN; Architectural Views of Software; FURPS+.*

Resumen:

La gestión eficiente de los flujos de trabajo es esencial para las clínicas de oncología, con el objetivo de garantizar la calidad de la atención y la eficacia de los tratamientos. Este trabajo propone una solución basada en arquitectura de flujo de trabajo para

DOI: https://doi.org/10.29327/2384439.3.3-6

optimizar los procesos clínicos en una clínica de oncología, con enfoque en la eficiencia operativa. La propuesta se basa en investigaciones documentales y revisiones bibliográficas de requisitos clínicos, así como en la definición de una arquitectura adaptada, utilizando técnicas de modelado para garantizar escalabilidad, seguridad y mantenibilidad. Se presentan diagramas con notaciones UML y BPMN como base para proyectos futuros, con el objetivo de unificar y automatizar los flujos de trabajo. Los resultados obtenidos cubren las dos primeras etapas del ciclo de vida arquitectónico (análisis de requisitos y diseño). Aunque esta propuesta no implica una implementación completa, proporciona un marco robusto para el desarrollo futuro de sistemas de software funcionales, cohesivos y mantenibles, con la necesidad de validaciones prácticas.

Palabras clave: *Workflow; BPMN; Vistas arquitectónicas de software; FURPS+.*

DOI: https://doi.org/10.29327/2384439.3.3-6

1. INTRODUÇÃO

Conforme apontado por Chaudhry et al. (2006), os Sistemas de Informação (SI) no Setor de Saúde oferecem inúmeras vantagens, incluindo acesso total aos dados do paciente, redução de erros, minimização de custos, segurança de dados e atendimento aprimorado ao paciente.

De acordo com os estudos de Gooch (2011), a aplicação de uma arquitetura workflow na área de saúde também contribui para a segurança dos pacientes e a conformidade com as regulamentações. A rastreabilidade das atividades, o registro de auditoria e a aplicação consistente de protocolos clínicos são elementos fundamentais nos cuidados.

No contexto das clínicas de oncologia, a gestão eficiente dos fluxos de trabalho é de vital importância para garantir a qualidade do atendimento e o sucesso dos tratamentos oferecidos aos pacientes. A proposta de um software baseado em arquitetura workflow surge como uma abordagem promissora para otimizar e aprimorar os processos clínicos nesse cenário complexo.

Segundo Winchester (1999), a arquitetura workflow é uma estrutura que permite a automação e o gerenciamento de fluxos de trabalho, definindo a sequência lógica de atividades e a interação entre diferentes sistemas e profissionais.

Martin (2019) pontua que é comum os programadores, frente a um novo desafio, partirem para o esforço de implementação sem um trabalho prévio de planejamento, colocando em evidência os comportamentos de uma solução em detrimento da sua arquitetura, consequentemente isso resulta em sistemas rígidos, caros e difíceis de manter, comprometendo assim sua longevidade, o oposto do que é desejado para um software que pode armazenar informações vitais.

A proposta de um software baseado em arquitetura workflow visa possibilitar a implementação de mecanismos de controle e acompanhamento, garantindo a conformidade com os padrões e a qualidade dos processos, reforçando o que Sommerville (2011) afirma sobre a arquitetura de um software.

Este trabalho tem como objetivo geral propor um software baseado em uma arquitetura de workflow para uma clínica de oncologia, com foco na otimização do fluxo de tratamento e acompanhamento dos pacientes. O estudo irá identificar as necessidades da clínica, analisando os desafios e as oportunidades de aplicar essa abordagem ao contexto oncológico. É importante ressaltar que este trabalho não contempla a implementação do sistema, mas abrange as duas primeiras fases do ciclo de vida arquitetural: Análise de Requisitos e Design da Arquitetura, incluindo a proposta de um projeto de arquitetura utilizando as tecnologias adequadas.

Por fim, este trabalho se justifica por contribuir para a agilização das práticas clínicas, oferecendo um embasamento sólido para a proposta de soluções tecnológicas que otimizem a gestão dos fluxos de trabalho, melhorem a eficiência dos processos e promovam um cuidado mais integrado e personalizado aos pacientes, refletindo, assim, a evolução da área da saúde com o suporte tecnológico apropriado.

DOI: https://doi.org/10.29327/2384439.3.3-6

2. Fundamentação Teórica

Conforme Pressman (2011), a arquitetura de software é a base de um sistema, envolvendo seus componentes, propriedades visíveis e interações. Ela abrange as principais decisões de design e organização que moldam o sistema, é responsável por garantir que ele atenda aos requisitos funcionais e não funcionais. No contexto de arquiteturas de software, a arquitetura workflow é uma abordagem específica que se concentra na automação e gerenciamento dos fluxos de trabalho dentro do sistema, facilitando a coordenação entre processos e melhorando a eficiência.

Para dar uma visão mais clara sobre a arquitetura de software, pode ser verificado:

> Você não tentaria construir uma casa sem uma planta, não é mesmo? Você também não desenharia as plantas começando pela distribuição dos encanamentos da casa. Deve-se partir do contexto geral — a casa em si — antes de se preocupar com os detalhes. É exatamente isso que faz o projeto da arquitetura — ele dá uma visão geral e garante que você o entendeu de forma correta (Pressman, 2011, p. 387-388).

A arquitetura de software não se baseia apenas em fazer diagramas, ela vai muito mais adiante do que se pode imaginar, desde a identificação de componentes, relacionamentos, camadas, padrões arquiteturais e qualidades arquiteturais até chegar à implementação e teste do software, gerando benefícios na utilização.

Larman (2007) enfatiza a importância da arquitetura de software como um elemento fundamental para o sucesso de um sistema. Ele destaca os benefícios da arquitetura adequada, como comunicação eficaz, reutilização, manutenibilidade, escalabilidade e testabilidade, e incentiva uma abordagem iterativa e flexível no projeto da arquitetura, como pode ser visto no Quadro 1.

Quadro 1 - Benefícios da arquitetura de software

Benefício	Definição
Compreensão e comunicação	A arquitetura fornece uma representação visual clara do sistema, o que não apenas facilita a compreensão dos componentes e suas interações, mas também serve como uma linguagem comum entre a equipe de desenvolvimento e os *stakeholders*. Isso promove uma colaboração mais eficaz, garantindo que todos os envolvidos tenham uma visão alinhada dos objetivos do projeto e das funcionalidades do sistema, reduzindo ambiguidades e mal-entendidos.
Reutilização	Identifica componentes reutilizáveis, economizando tempo e esforço no desenvolvimento.
Manutenibilidade	Uma boa arquitetura facilita a manutenção e evolução do sistema ao permitir que as equipes identifiquem rapidamente a origem de problemas e implementem correções ou melhorias. Isso se dá por meio de uma estrutura modular que separa as funcionalidades, o que simplifica a realização de alterações e a adição de novos recursos. Essa flexibilidade é essencial em um ambiente de tecnologia em constante mudança, onde requisitos e demandas dos usuários podem evoluir rapidamente.
Escalabilidade	Permite dimensionar o sistema para lidar com maior carga ou novos requisitos de desempenho.
Testabilidade	Facilita a criação de testes, melhorando a cobertura e detecção de defeitos.

Fonte: Adaptado de (Larman, 2007)

DOI: https://doi.org/10.29327/2384439.3.3-6

2.1 Arquitetura Workflow

Segundo Aalst (2004), workflow é uma representação formalizada e automatizada de um processo de negócio que define a sequência de atividades, a lógica de controle e a interação entre pessoas, sistemas e recursos, visando alcançar um objetivo específico. Ele envolve a coordenação e o gerenciamento eficiente das tarefas, informações e recursos envolvidos no processo, garantindo a execução ordenada e consistente das atividades, com base em regras de negócio e políticas predefinidas. Sommerville (2011) ainda afirma a importância de modelar corretamente os fluxos de trabalho, levando em consideração os diferentes tipos de tarefas, o sequenciamento das atividades, a coordenação entre os participantes e a integração com outros sistemas. Ele destaca que a arquitetura de workflow deve ser projetada para suportar as demandas específicas do ambiente de negócios.

Conforme Aalst (2004) aponta, a arquitetura workflow envolve a automação, gerenciamento e coordenação dos processos de negócio por meio de sistemas de informação. Ela se baseia em uma compreensão clara dos processos de negócio e utiliza técnicas de modelagem para representá-los visual ou textualmente. Esses modelos permitem capturar os aspectos estruturais e comportamentais dos processos, definindo as atividades envolvidas e suas interações, como sugere o Quadro 2.

Quadro 2 - Conceitos da arquitetura *workflow*

Processo de negócio	Série de atividades coordenadas para atingir um objetivo específico, base da arquitetura *workflow*.
Modelagem de processo	Representação de processos usando notações como BPMN para capturar aspectos estruturais e comportamentais.
Workflow	Automação e gerenciamento de processos por sistemas, com foco em projetar fluxos de trabalho flexíveis e eficazes.
Coordenação e colaboração	Coordena pessoas, sistemas e recursos em processos, com atribuição de tarefas e interação entre participantes.
Gerenciamento de exceções	Aborda exceções e desvios do fluxo normal, com técnicas como roteamento dinâmico e tratamento de exceções.
Monitoramento e análise	Fornece mecanismos para monitorar a execução dos processos e coletar dados para análise.

Fonte: Adaptado de (Aalst, 2004)

2.2 Ciclo de vida arquitetural

O ciclo de vida arquitetural refere-se ao processo pelo qual a arquitetura de um sistema de software é concebida, projetada, implementada e evoluída ao longo do tempo. Segundo Bass, Clements e Kazman (2012), o ciclo de vida arquitetural geralmente abrange várias etapas, como é exibido no Quadro 3.

Para Bass, Clements e Kazman (2012), este é um processo contínuo e iterativo, pois a arquitetura precisa acompanhar as mudanças no ambiente, nos requisitos e nas tecnologias. É essencial para garantir a robustez, flexibilidade e adaptabilidade do sistema de *software* ao longo do tempo.

DOI: https://doi.org/10.29327/2384439.3.3-6

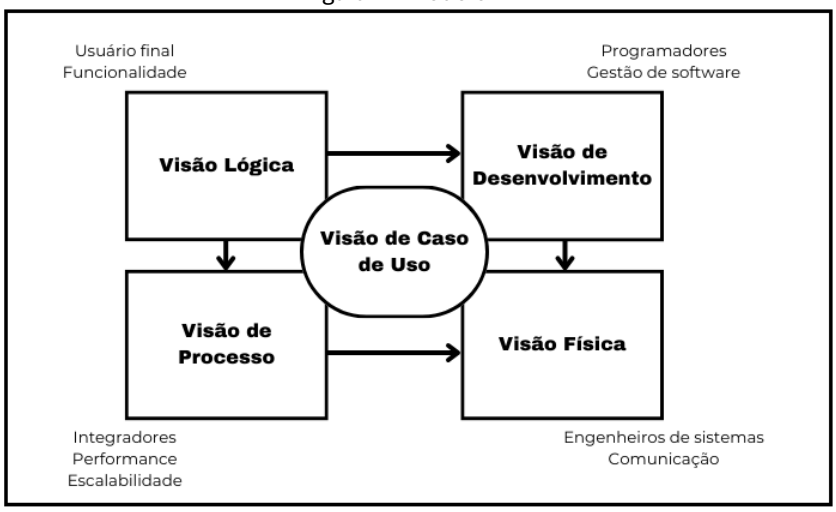
Quadro 3 - Etapas do ciclo de vida e suas definições

Análise de requisitos	Coleta e analisa RF e RNF, considerando as necessidades dos usuários e restrições técnicas.
Design arquitetural	Define a estrutura do sistema.
Implementação	Codificação dos componentes do sistema com base no *design* arquitetural e tecnologias selecionadas.
Teste e validação	Testes de funcionalidade, desempenho e segurança para garantir que o sistema atenda aos requisitos.
Implantação e operação	Instalação e configuração do sistema em produção, tornando-o operacional e disponível para uso.
Evolução e manutenção	Ajustes e melhorias na arquitetura para atender novos requisitos e resolver problemas identificados.

Fonte: Adaptado de (Bass, Clements e Kazman, 2012)

De acordo com Kruchten (1995), a descrição do funcionamento de sistemas de *software* é fundamentada no uso de múltiplas visões concorrentes. Essas visões são empregadas para apresentar o sistema sob diversas perspectivas, incluindo as dos usuários finais, desenvolvedores e gerentes de projetos. O modelo 4+1 é composto por quatro visões principais: a visão lógica, que descreve a estrutura do sistema em termos de classes e objetos; a visão de desenvolvimento, que foca na organização do código e nos componentes de *software*; a visão de processo, que aborda a dinâmica do sistema, incluindo suas interações e fluxos de trabalho; e a visão física, que representa a infraestrutura e a distribuição dos componentes em *hardware*. A quinta visão, chamada de **visão de casos de uso** ou "+1", ilustra a arquitetura do sistema por meio de cenários que integram as demais visões. A Figura 1 apresenta um exemplo deste modelo, que será adotado ao longo deste trabalho para guiar a análise e o desenvolvimento da arquitetura proposta.

Figura 1 - Modelo 4+1



Fonte: Adaptado de (Kruchten, 1995)

3. MÉTODO

Diante do contexto das clínicas de oncologia e da necessidade de aprimorar a eficiência dos processos clínicos, este estudo visa auxiliar na organização e agilização nos processos de tratamento e acompanhamento de pacientes, com base na arquitetura de fluxo de trabalho. Foram realizadas pesquisas

DOI: https://doi.org/10.29327/2384439.3.3-6

bibliográficas e documentais para fundamentar a proposta.

3.1 Ciclo de vida arquitetural

O estudo de caso constitui-se de uma clínica oncológica que precisa de uma aplicação de software para ter um melhor fluxo de trabalho, como, controlar a chegada de pacientes, fazer um registro dos pacientes, consultas, profissionais de saúde, quimioterapias e também das evoluções.

3.2 Ciclo de vida arquitetural

O Ciclo de Vida Arquitetural é um processo fundamental que abrange as fases de desenvolvimento de um sistema, desde a identificação de requisitos até a implementação e manutenção, como mencionado na seção 2.2. Neste trabalho, a fase de Análise de Requisitos é essencial para compreender as necessidades da clínica e servir como base para a proposta. Para entender as necessidades da clínica e elaborar a proposta, foram adotadas duas abordagens principais:

Revisão Bibliográfica: Foi feita uma revisão da literatura sobre gestão de fluxos de trabalho em clínicas de oncologia e arquitetura workflow, que serviu como base teórica para a proposta.

Pesquisa Documental: Foram realizadas pesquisas em artigos *online*, selecionados para fornecer uma compreensão mais detalhada dos processos e das demandas das clínicas, sendo eles (Tummers et al., 2021; Bulcão-Neto et al., 2022; Pufahl et al., 2022; De Ramón Fernández et al., 2020; Chaudhry et al., 2006).

Após a análise do estudo de caso, foram identificados os requisitos funcionais e não funcionais. Os requisitos não funcionais (RNF) incluem testabilidade, usabilidade, interoperabilidade, manutenibilidade, escalabilidade, segurança e desempenho.

No Quadro 4, são descritos os requisitos funcionais (RF), que definem as funcionalidades e ações que o sistema deverá realizar para atender às necessidades operacionais. Esses requisitos abrangem desde o gerenciamento de dados cadastrais até a exibição de informações e o controle de atividades relacionadas ao atendimento médico e procedimentos específicos.

Quadro 4 - Requisitos Funcionais (RF)

Código	Descrição
RF01	Manter as informações cadastrais dos pacientes.
RF02	Visualizar uma agenda com as consultas.
RF03	**Confirmar a chegada do paciente.**
RF04	Registrar os dados cadastrais de consultas e evoluções.
RF05	**Listar as evoluções e os pacientes cadastrados.**
RF06	Visualizar as evoluções de um paciente específico.
RF07	Registrar procedimentos de quimioterapia.
RF08	**Solicitar medicamentos ao posto de enfermagem.**
RF09	Manter as informações dos profissionais de saúde.
RF10	Exibir a agenda com consultas e quimioterapias agendadas para o dia.

Fonte: Autores (2024)

DOI: https://doi.org/10.29327/2384439.3.3-6

Em seguida, no Quadro 5, são apresentadas as **regras de negócio (RN)**, que estabelecem as políticas e restrições que o sistema deverá obedecer para garantir a correta execução dos processos. Essas regras são fundamentais para assegurar o alinhamento do sistema com as normas operacionais e práticas da organização.

Quadro 5 - Regras de Negócio (RN)

Código	Descrição
RN01	Proíbe o registro de consultas em datas e horários que já estejam ocupados.
RN02	Impede o agendamento de novas quimioterapias em horários já reservados.
RN03	Remove o agendamento da agenda após 15 minutos de atraso.
RN04	Limita os tipos de evolução permitidos a: Intercorrência médica ou de enfermagem.
RN05	Aceita apenas convênios como forma de pagamento, outros métodos devem ser tratados externamente.

Fonte: Autores (2024)

3.3 Ciclo de vida arquitetural

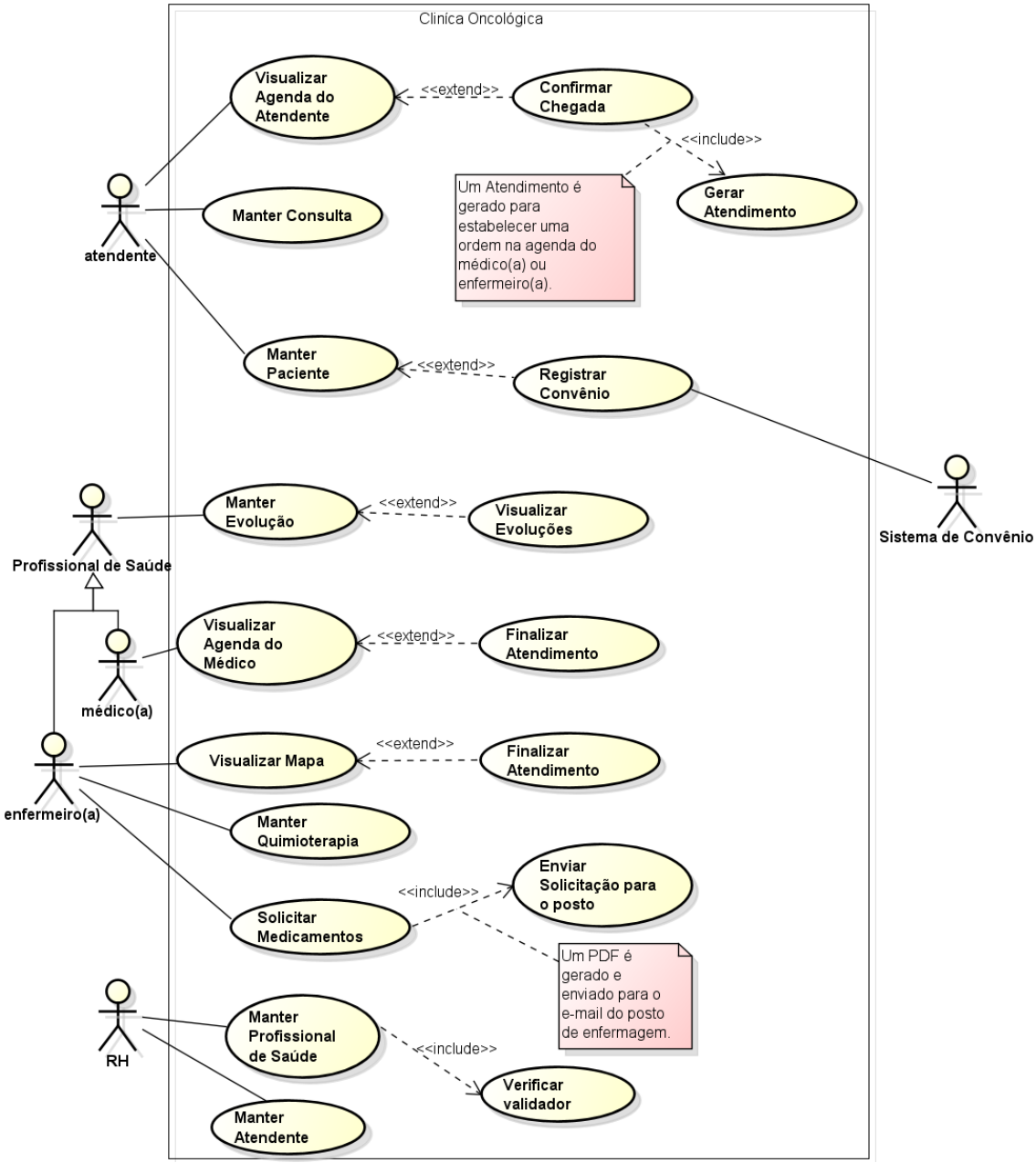
A Visão de Cenário, proposta por Kruchten (1995), utiliza casos de uso para demonstrar como o sistema atende a requisitos funcionais e não funcionais, validando a arquitetura em relação às necessidades dos usuários. Essa abordagem permite uma análise mais aprofundada de como diferentes componentes interagem sob condições reais de uso, garantindo que a arquitetura projetada seja robusta e flexível o suficiente para suportar mudanças. Os cenários também oferecem uma visão prática e iterativa da evolução do sistema, permitindo que stakeholders visualizem o sistema em operação, ajudando a identificar potenciais problemas antes que se tornem críticos. Além disso, a modelagem com cenários auxilia na identificação de requisitos ocultos e na priorização de funcionalidades essenciais. Booch, Jacobson e Rumbaugh (2005) afirmam que o diagrama de caso de uso é uma representação gráfica que detalha a interação entre usuários e o sistema, facilitando a comunicação e a compreensão dos requisitos. Essa visualização é fundamental para alinhar a visão dos desenvolvedores com as expectativas dos usuários e garantir que todos os envolvidos compartilhem um entendimento claro do escopo do sistema. No caso deste trabalho, a Figura 2 ilustra essas interações de maneira que está diretamente alinhada com os objetivos e requisitos discutidos, além de servir como uma ferramenta para validação contínua ao longo do ciclo de vida do desenvolvimento.

No estudo, foram identificados vários casos de uso, cada um refletindo diferentes aspectos funcionais e não funcionais do sistema. No entanto, apenas um deles será apresentado neste trabalho devido a restrições de espaço. A escolha desse caso específico foi feita com base em sua relevância para os objetivos do projeto e sua capacidade de ilustrar as interações essenciais entre usuários e sistema. A aplicação do modelo FURPS (+) seguiu a abordagem de Eeles (2005) para ajudar a garantir que todos os aspectos relevantes sejam considerados durante o desenvolvimento. Essa abordagem estruturada permitirá uma avaliação abrangente dos requisitos, facilitando a identificação de potenciais riscos e oportunidades de melhoria. A análise detalhada pode ser observada no Quadro 6, que apresenta uma visão clara dos critérios considerados e como eles se

DOI: https://doi.org/10.29327/2384439.3.3-6

relacionam com o caso de uso selecionado.

Figura 2 - Diagrama de Caso de Uso



Fonte: Autores (2024)

Embora o Quadro 6 reconheça determinados atributos de qualidade como requisitos não funcionais, é importante ressaltar que outros atributos foram identificados nos cenários arquiteturais dos demais casos de uso.

DOI: https://doi.org/10.29327/2384439.3.3-6

Quadro 6 - Caso de Uso Manter Evolução

Caso de Uso: CSU04: Manter Evolução		
Descrição: O profissional de saúde poderá fazer o registro de uma evolução no sistema.		
Ator Primário: Médico, Enfermeiro		
Ator(es) Secundário(s): N/A.		
Precondições: Estar logado no sistema.		
Fluxo Principal: 1. O Médico insere a data que a evolução foi realizada, o tipo de evolução (intercorrência, médica ou enfermagem, conforme a RN04), o paciente que recebeu a evolução, o profissional de saúde que realizou a evolução e o procedimento que foi feito. 2. O sistema verifica se há médicos e pacientes com os dados que foram preenchidos. 3. O sistema confirma que existem os dados já registrados no sistema e registra a evolução.		
Fluxo Alternativo (3): Paciente ou médico não encontrado. a) O sistema não encontra um paciente ou médico com os campos preenchidos. Retorna ao passo 1. Fluxo Alternativo (1): Consultar evoluções ou evolução. a) O Médico irá selecionar a opção de consultar evolução. b) O sistema irá trazer as opções de consultar uma lista de evolução o uma evolução em específica. c) O médico poderá fazer alterações ou excluir a evolução que foi aberta ou apresentada.		
Pós-condições: A evolução foi registrada.		
Regras de Negócio Relacionadas: RN04		
Cenários Arquiteturais	Requisito Arquitetural (conforme FURPS+)	Descrição do Requisito Arquitetural
Para todo o caso de uso	Funcionalidade/Segurança (Restrição de Acesso)	O usuário precisa ser autenticado pelo sistema.
1. b)	Performance (Desempenho)/ Tempo de Resposta	O sistema deve exibir o resultado no máximo em 5 segundos.
Para todo o caso de uso	Usabilidade/ Interface intuitiva	O Sistema deve apresentar campos organizados e funções já pré estabelecidas para o usuário.

Fonte: Autores (2024)

3.4 Ciclo de vida arquitetural

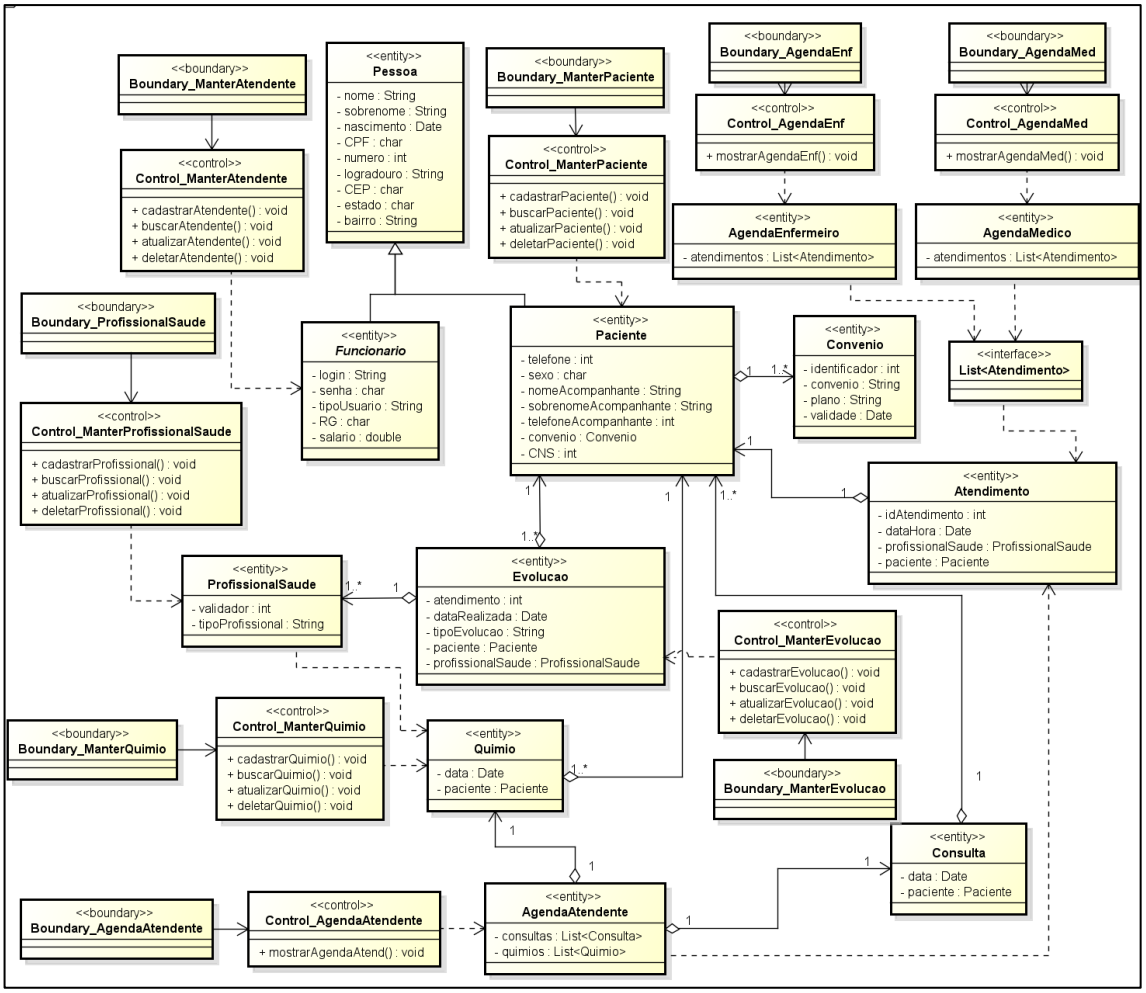
O Ciclo Arquitetural é um conjunto estruturado de etapas que orienta o desenvolvimento de sistemas, destacando a fase de *Design* Arquitetural como uma das mais críticas. Essa fase foca na definição da estrutura do sistema, levando em consideração as interações entre os diversos componentes e as necessidades dos usuários. Durante o *Design* Arquitetural, é essencial criar representações que visualizem a organização do *software*, suas funcionalidades e suas inter-relações, facilitando a compreensão da arquitetura proposta. Além disso, essas representações ajudam a identificar potenciais problemas de integração e a validar decisões arquiteturais, assegurando que a solução atenda aos requisitos funcionais e não funcionais. Assim, essa fase é fundamental para garantir um alinhamento eficaz entre as expectativas dos *stakeholders* e a implementação do sistema.

DOI: https://doi.org/10.29327/2384439.3.3-6

3.4.1 Visão Lógica

A Visão Lógica, conforme proposta por Kruchten (1995), concentra-se na representação dos componentes do sistema e nas interações entre eles. Ela detalha a lógica de negócio, utilizando diagramas de classes e objetos para ilustrar a estrutura e a organização do software em um nível conceitual. Essa visão é fundamental para entender como os elementos do sistema se conectam e colaboram para atender aos requisitos funcionais. No caso deste trabalho, a Figura 3 apresenta o diagrama de classes que representa a visão lógica da arquitetura proposta para o sistema de software destinado à clínica oncológica.

Figura 3 - Diagrama de Classes com categorização BCE



Fonte: Autores (2024)

3.4.2 Visão de Desenvolvimento

A Visão de Desenvolvimento, conforme descrito no modelo 4+1 de Kruchten (1995), enfoca a organização do software, mostrando como os componentes e pacotes de código são agrupados em módulos e bibliotecas. Essa visão é importante para entender a estrutura do código, facilitando a manutenção e a evolução do sistema. Ela ajuda os desenvolvedores a gerenciar o código de maneira mais eficaz, promovendo práticas de desenvolvimento eficientes.

DOI: https://doi.org/10.29327/2384439.3.3-6

O diagrama de pacotes é uma ferramenta essencial na modelagem de sistemas de software, conforme destacado por Larman (2007). Ele organiza pacotes, camadas ou componentes, proporcionando uma visão clara da estrutura lógica do sistema e das relações entre suas partes. Essa organização não apenas melhora a comunicação entre os membros da equipe, mas também facilita o entendimento da arquitetura durante o desenvolvimento, permitindo que todos tenham uma compreensão compartilhada das interações e dependências. Na Figura 4, o diagrama de pacotes elaborado para este trabalho ilustra as camadas do sistema, incluindo a lógica de negócio, a camada de informação e a de apresentação. Ele representa as relações entre os pacotes, destacando como eles podem interagir de forma independente do código, o que contribui para uma maior flexibilidade e manutenibilidade do sistema. Essa abordagem permite que alterações em um pacote sejam feitas sem afetar diretamente os demais, promovendo uma arquitetura mais robusta e escalável. Além disso, a utilização de diagramas de pacotes facilita a identificação de pontos de integração e dependência, o que é crucial para a implementação de novas funcionalidades e para a identificação de potenciais áreas de melhoria dentro da arquitetura existente. Essa clareza na estrutura do sistema é vital para garantir a eficácia do desenvolvimento ágil e a adaptação a mudanças de requisitos ao longo do ciclo de vida do software.

3.4.3 Visão de Processo

A Visão de Processo, também parte do modelo 4+1 Kruchten (1995), aborda a dinâmica do sistema, focando em como os processos interagem e se comportam ao longo do tempo. Essa visão utiliza diagramas de atividades e fluxos de controle para descrever o fluxo de informações e a sincronização entre diferentes processos. É essencial para compreender a execução do sistema e garantir que as interações ocorram conforme esperado.

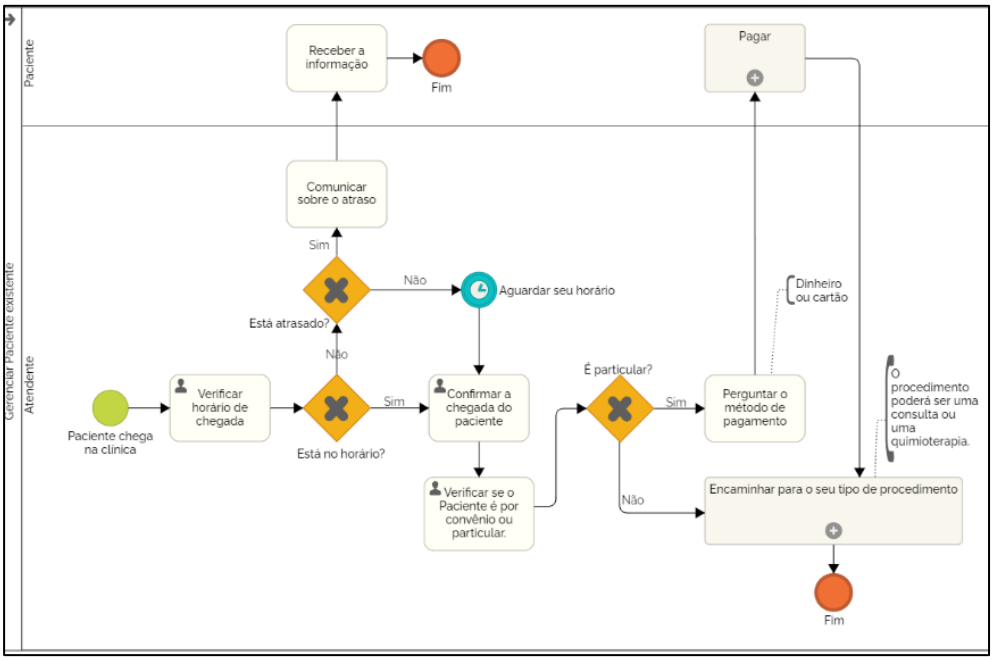
A modelagem BPMN (*Business Process Model and Notation*) é uma notação gráfica projetada para representar processos de negócios de maneira clara e compreensível. Ela utiliza uma variedade de símbolos e convenções que descrevem elementos essenciais, como atividades, eventos, fluxos, decisões e interações. Essa ferramenta é amplamente utilizada na análise, automação e comunicação de processos, permitindo que todos os envolvidos tenham uma visão compartilhada e precisa do fluxo de trabalho. A notação inclui componentes como atividades, eventos, fluxos de sequência, decisões, *gateways*, *pools* e *lanes*, conforme destacado por Silver (2012). Essa estruturação facilita a documentação e o entendimento dos processos, promovendo eficiência e alinhamento entre as equipes.

Serão apresentados dois diagramas BPMN que ilustram o processo de atendimento ao paciente, essenciais para entender a dinâmica do caso de uso "Manter Evolução". Esses diagramas fornecem uma representação visual clara das etapas envolvidas no atendimento, permitindo que os *stakeholders* visualizem como as interações ocorrem em tempo real. A Figura 5, em particular,

DOI: https://doi.org/10.29327/2384439.3.3-6

demonstra o momento da chegada de um paciente registrado, destacando os eventos e atividades que marcam o início do processo.

Figura 4 - Diagrama BPMN para o Caso de Uso Manter Evolução

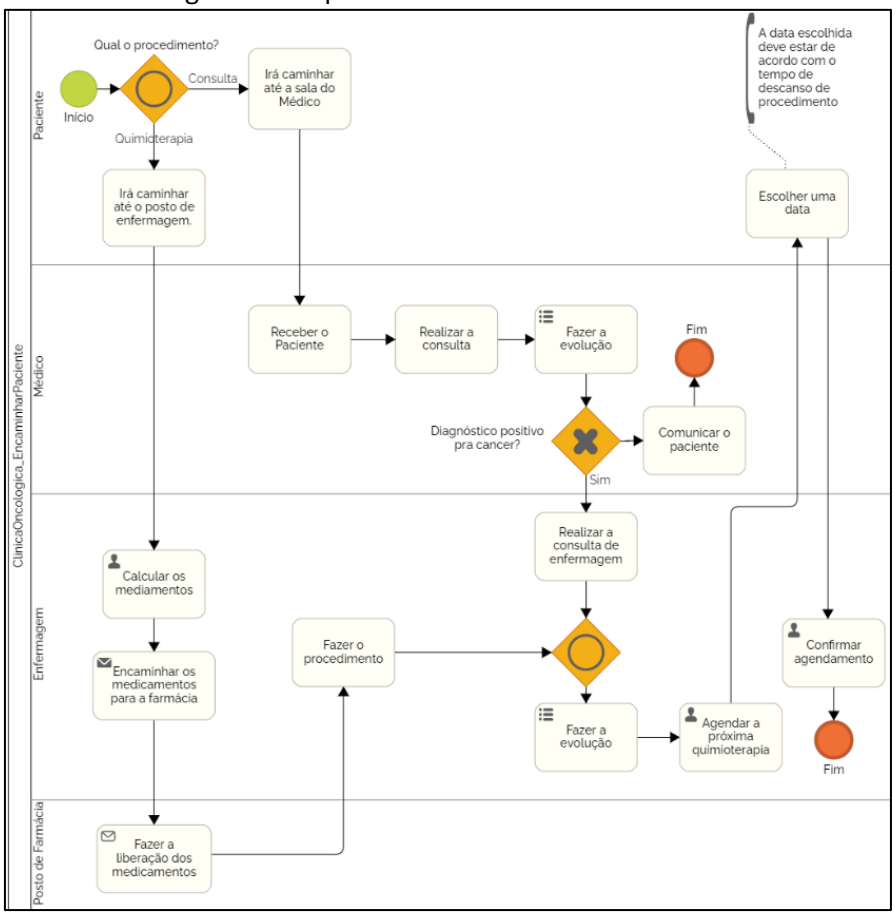


Fonte: Autores (2024)

Após a chegada, o diagrama da Figura 6 detalha o subprocesso de encaminhamento do paciente, abordando as atividades necessárias para determinar o tipo de procedimento mais adequado, levando em consideração as necessidades e condições específicas do paciente. Essa abordagem assegura um atendimento personalizado e eficiente.

DOI: https://doi.org/10.29327/2384439.3.3-6

Figura 5 - Subprocesso de encaminhamento



Fonte: Autores (2024)

3.4.4 Visão Física

A Visão Física, conforme proposta por Kruchten (1995), representa a infraestrutura do sistema, detalhando a disposição dos componentes de hardware e suas configurações de rede. Essa visão é crucial para a implementação e operação do sistema em ambientes reais, proporcionando uma compreensão clara de como os elementos físicos se conectam e interagem para suportar as funcionalidades do software.

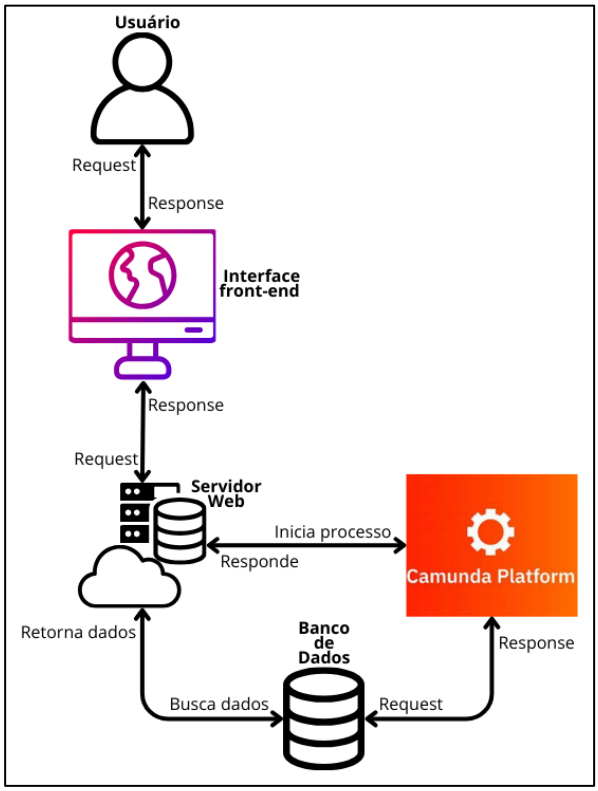
No contexto da proposta de uma solução baseada em arquitetura workflow para a clínica oncológica, a arquitetura de aplicação web é estruturada em três camadas: a Camada de Apresentação (Front-end), que lida com a interface do usuário, a Camada de Aplicação (Back-end), responsável pela lógica de negócio, e a Camada de Dados (Banco de Dados), onde as informações são armazenadas e gerenciadas. A seleção das ferramentas e tecnologias apropriadas para cada camada é essencial, considerando a integração com diferentes linguagens e APIs, e dependerá dos requisitos específicos da clínica e da disponibilidade das soluções.

A Figura 7 ilustra a implementação da parte física da arquitetura proposta, destacando a disposição e interconexão dos componentes nas diferentes camadas. Nesta representação, nota-se a utilização da ferramenta Camunda, uma plataforma de código aberto para automação de processos de negócios.

DOI: https://doi.org/10.29327/2384439.3.3-6

Segundo a Camunda (2024), essa ferramenta é eficaz na orquestração do fluxo de trabalho entre as camadas, garantindo que os processos de tratamento dos pacientes sejam coordenados de forma integrada e eficiente.

Figura 6 - Arquitetura Web



Fonte: Autores (2024)

4. Resultados e Discussão

Os resultados obtidos até o momento referem-se às duas primeiras etapas do ciclo de vida arquitetural: análise de requisitos e design arquitetural. A análise de requisitos foi crucial para identificar as necessidades específicas da clínica oncológica, abrangendo requisitos funcionais e não funcionais. O design arquitetural traduziu essas necessidades em uma estrutura técnica coerente, definindo as interações entre os componentes do sistema de forma alinhada às diretrizes de qualidade apresentadas pelos autores. Entretanto, ainda não foi possível validar esses resultados com a implementação prática, limitando a análise a uma proposta teórica.

As quatro etapas restantes do ciclo de vida arquitetural: implementação, teste, implantação e manutenção, representam resultados esperados essenciais para a concretização do projeto. A implementação consistirá na codificação dos componentes do sistema, onde as decisões de design serão colocadas em prática. Os testes subsequentes, incluindo testes de unidade, integração, sistema e aceitação, serão fundamentais para validar a qualidade do sistema e garantir que todos os requisitos sejam atendidos. Este processo de testes permitirá a identificação de problemas precoces e a realização de ajustes, podendo até levar

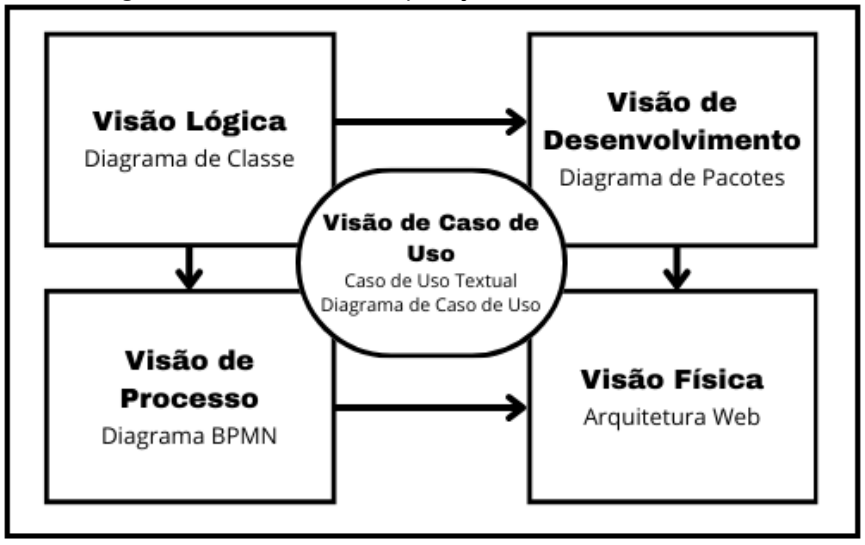
DOI: https://doi.org/10.29327/2384439.3.3-6

a iterações entre as etapas do ciclo arquitetural, conforme necessário.

A fase de implantação permitirá que o sistema seja disponibilizado aos usuários finais, enquanto a fase de manutenção garantirá o aprimoramento contínuo do sistema com base no feedback dos usuários e nas mudanças nas necessidades da clínica. Essa abordagem iterativa e colaborativa assegura que a solução final seja robusta, escalável e alinhada às demandas da clínica, proporcionando um fluxo de trabalho otimizado e um acompanhamento eficaz dos pacientes.

Na Figura 8, o modelo 4+1 foi replicado, considerando os elementos presentes neste trabalho que compõem cada visão arquitetural.

Figura 7 - Modelo 4+1 com aplicação dos conceitos utilizados



Fonte: Autores (2024)

5. Considerações Finais

O trabalho focou nas duas primeiras etapas do ciclo de vida arquitetural, analisando os requisitos e propondo o *design* arquitetural para uma clínica oncológica. Essas etapas foram concluídas com sucesso, resultando em uma estrutura inicial robusta que atende às necessidades identificadas da clínica. As etapas seguintes, que envolvem implementação, testes, implantação e manutenção, estão planejadas como resultados futuros. A implementação completa e os testes práticos ainda não foram realizados, o que limita a avaliação da eficácia final do sistema. Em relação ao objetivo do trabalho, este foi atingido, evidenciando resultados que confirmam essa conclusão.

Embora tenha sido capaz de propor uma arquitetura detalhada e consistente, a falta de implementação prática significa que ainda não é possível validar completamente a eficiência operacional. Para confirmar a efetividade do sistema, a implementação precisa ser feita, seguida de uma fase de testes que valide os resultados previstos.

Trabalhos futuros podem focar na implementação do sistema proposto, realizando testes de usabilidade e de desempenho para validar a eficácia do

DOI: https://doi.org/10.29327/2384439.3.3-6

workflow na prática. Além disso, a avaliação do impacto no tempo de atendimento e na qualidade do cuidado ao paciente deverá ser parte de uma investigação mais aprofundada após a implementação.

Entre as limitações identificadas, destacam-se a necessidade de integração com sistemas legados de prontuários eletrônicos, o que pode exigir um esforço significativo de adaptação. Além disso, a dependência de treinamento da equipe clínica para operar a nova arquitetura pode representar um desafio em termos de adoção da tecnologia.

REFERÊNCIAS

AALST, W. V. D. **Workflow Management: Models, Methods, and Systems**. Cambridge: The MIT Press, 2004.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice**. 3. ed. New Jersey: Person Education, 2012.

BOOCH, G.; JACOBSON, G.; RUMBAUGH, J. **UML: Guia do usuário**. 2. ed. Rio de Janeiro: Elsevier, v. I, 2005.

BULCÃO-NETO, R. F.; NETO, V. V. G.; MACEDO, A. A. **A reference architecture for healthcare systems with coded terminology support**. In: *Intermountain Engineering, Technology and Computing (IETC)*, 2022, Orem, UT, USA. Anais... 2022. p. 1-6. DOI: 10.1109/IETC54973.2022.9796889.

CAMUNDA. *Camunda Platform.* Disponível em: https://camunda.com. Acesso em: 5 nov. 2024.

CHAUDHRY, B.; WANG, J.; WU, S.; MAGLIONE, M.; MOJICA, W.; ROTH, E.; MORTON, S.; SHEKELLE, P. **Systematic review: impact of health information technology on quality, efficiency, and costs of medical care**. *Annals of Internal Medicine*, v. 144, n. 10, p. 742-752, 2006. DOI: 10.7326/0003-4819-144-10-200605160-00125.

DE RAMÓN FERNÁNDEZ, A.; RUIZ FERNÁNDEZ, D.; SABUCO GARCÍA, Y. **Business process management for optimizing clinical processes: a systematic literature review**. *Health Informatics Journal*, v. 26, n. 2, p. 1305-1320, 2020. DOI: 10.1177/1460458219877092.

EELES, P. **Capturing architectural requirements**. *IBM Rational Developer Works*, 2001. Disponível em: https://www.researchgate.net/publication/329760910_Capturing_Architectural_Requirements. Acesso em: 20 jun. 2023.

GOOCH, P.; ROUDSARI, A. **Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems**. *Journal of the American Medical Informatics Association (JAMIA)*, v. 18, n. 6, 2011. Disponível em: https://doi.org/10.1136/amiajnl-2010-000033. Acesso em: 16 out. 2024.

JACOBSON, I. **Object-Oriented Software Engineering: A Use Case Driven Approach**. Boston: Addison-Wesley, 1992.

KRUCHTEN, P. **The 4+1 View Model of Architecture**. *ResearchGate*, nov. 1995. Disponível em: https://www.researchgate.net/publication/220018231_The_41_View_Model_of_Architecture. Acesso em: 24 out. 2024.

LARMAN, C. **Utilizando UML e Padrões**. 3. ed. Porto Alegre: Bookman, 2007.

MARTIN, R. C. **Arquitetura Limpa - O Guia do Artesão para Estrutura e Design de Software**. 1. ed. Rio de Janeiro: Alta Books, 2019.

PRESSMAN, R. **Engenharia de Software: Uma Abordagem Profissional**. 9. ed. Porto Alegre: AMGH, 2011.

DOI: https://doi.org/10.29327/2384439.3.3-6

PUFAHL, L.; ZERBATO, F.; WEBER, B.; WEBER, I. **BPMN in healthcare: challenges and best practices**. *Information Systems*, v. 107, 2022, p. 102013. DOI: 10.1016/j.is.2022.102013.

SILVER, B. **BPMN 2.0 Handbook: Methods, Concepts, Case Studies and Standards**. 1. ed. Florida: Layna Fischer, 2012.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Person, 2011.

TUMMERS, J.; TOBI, H.; CATAL, C. **Designing a reference architecture for health information systems**. *BMC Medical Informatics and Decision Making*, v. 21, n. 210, 2021. Disponível em: https://doi.org/10.1186/s12911-021-01570-2. Acesso em: 27 out. 2024.

WINCHESTER. **Workflow patterns**. 1999. Disponível em: <www.workflowpatterns.com/documentation/documents/TC-1011_term_glossary_v3.pdf>. Acesso em: 28 maio 2023.

"Os conteúdos expressos no trabalho, assim como os direitos autorais de figuras e dados, bem como sua revisão ortográfica e das normas são de inteira responsabilidade do(s) autor(es)."

“O(s) autor(es) do trabalho declara(m) que durante a preparação do manuscrito foi(foram) utilizado(as) a(s) ferramenta(s)/serviço(s) [*GPT4o, Co-pilot e Consensus*] de Inteligência Artificial (IA) para [*analisar e sugerir possíveis correções gramaticais, e auxiliar nas pesquisas acadêmicas*]. Após utilizar esta ferramenta/serviço, os autores editaram e revisaram o conteúdo conforme necessário e assumem total responsabilidade pelo conteúdo da publicação.”