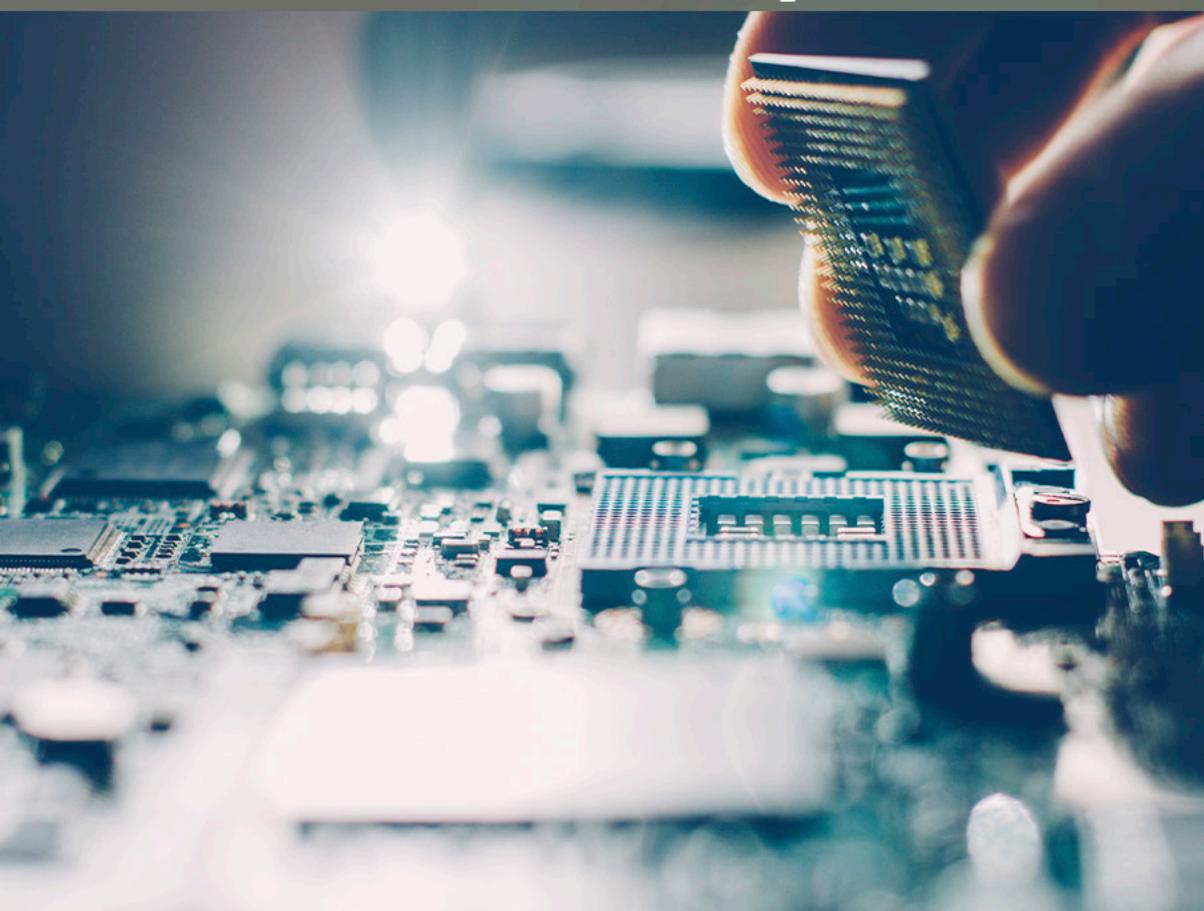


COLEÇÃO

DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO 3

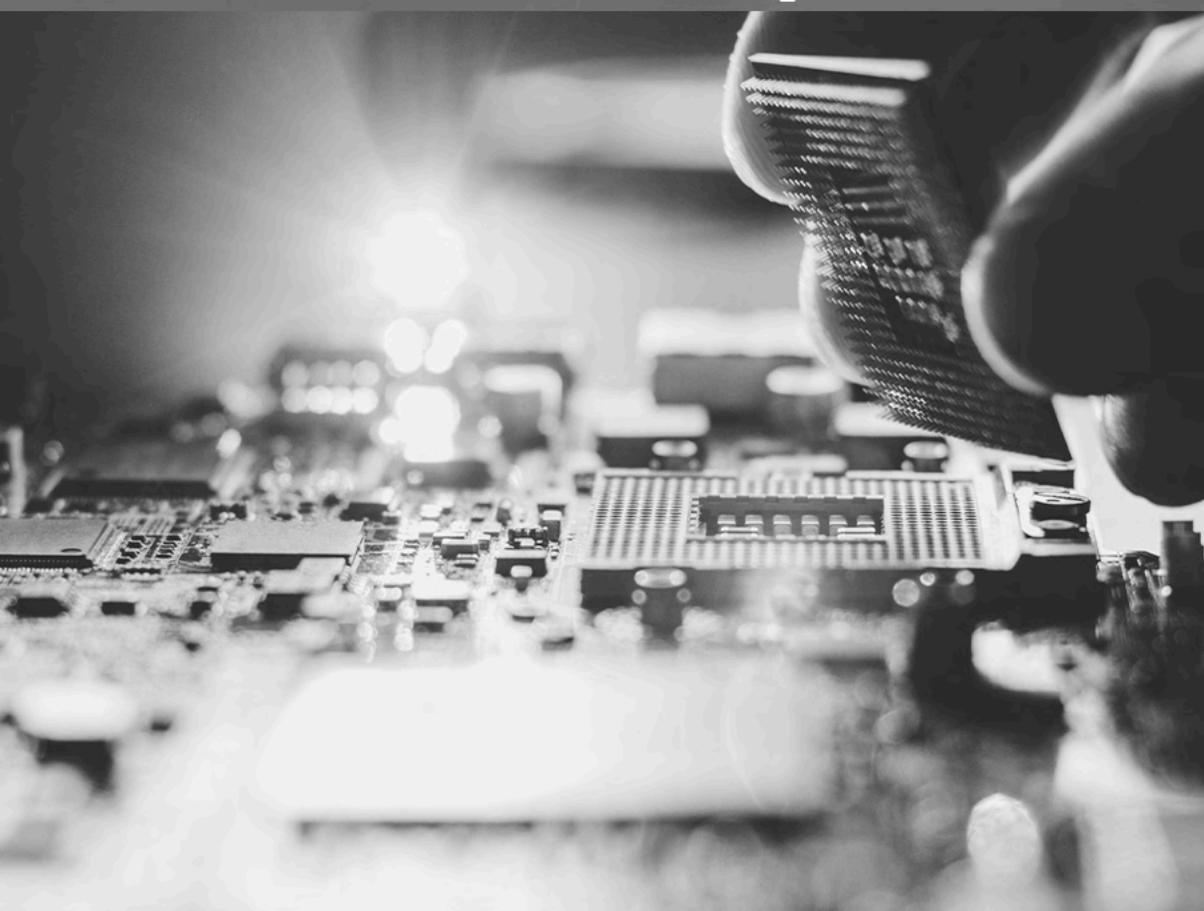


LILIAN COELHO DE FREITAS
(ORGANIZADORA)

Atena
Editora
Ano 2021

COLEÇÃO
DESAFIOS
DAS
ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO 3



LILIAN COELHO DE FREITAS
(ORGANIZADORA)

Atena
Editora
Ano 2021

Editora chefe

Profª Drª Antonella Carvalho de Oliveira

Editora executiva

Natalia Oliveira

Assistente editorial

Flávia Roberta Barão

Bibliotecária

Janaina Ramos

Projeto gráfico

Camila Alves de Cremo

Daphynny Pamplona

Gabriel Motomu Teshima

Luiza Alves Batista

Natália Sandrini de Azevedo

Imagens da capa

iStock

Edição de arte

Luiza Alves Batista

2021 by Atena Editora

Copyright © Atena Editora

Copyright do texto © 2021 Os autores

Copyright da edição © 2021 Atena Editora

Direitos para esta edição cedidos à Atena Editora pelos autores.

Open access publication by Atena Editora



Todo o conteúdo deste livro está licenciado sob uma Licença de Atribuição *Creative Commons*. Atribuição-Não-Comercial-NãoDerivativos 4.0 Internacional (CC BY-NC-ND 4.0).

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores, inclusive não representam necessariamente a posição oficial da Atena Editora. Permitido o *download* da obra e o compartilhamento desde que sejam atribuídos créditos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

Todos os manuscritos foram previamente submetidos à avaliação cega pelos pares, membros do Conselho Editorial desta Editora, tendo sido aprovados para a publicação com base em critérios de neutralidade e imparcialidade acadêmica.

A Atena Editora é comprometida em garantir a integridade editorial em todas as etapas do processo de publicação, evitando plágio, dados ou resultados fraudulentos e impedindo que interesses financeiros comprometam os padrões éticos da publicação. Situações suspeitas de má conduta científica serão investigadas sob o mais alto padrão de rigor acadêmico e ético.

Conselho Editorial

Ciências Exatas e da Terra e Engenharias

Prof. Dr. Adélio Alcino Sampaio Castro Machado – Universidade do Porto

Profª Drª Ana Grasielle Dionísio Corrêa – Universidade Presbiteriana Mackenzie

Prof. Dr. Carlos Eduardo Sanches de Andrade – Universidade Federal de Goiás

Profª Drª Carmen Lúcia Voigt – Universidade Norte do Paraná

Prof. Dr. Cleiseano Emanuel da Silva Paniagua – Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Prof. Dr. Douglas Gonçalves da Silva – Universidade Estadual do Sudoeste da Bahia
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná
Profª Drª Érica de Melo Azevedo – Instituto Federal do Rio de Janeiro
Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará
Profª Dra. Jéssica Verger Nardeli – Universidade Estadual Paulista Júlio de Mesquita Filho
Prof. Dr. Juliano Carlo Rufino de Freitas – Universidade Federal de Campina Grande
Profª Drª Luciana do Nascimento Mendes – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Prof. Dr. Marcelo Marques – Universidade Estadual de Maringá
Prof. Dr. Marco Aurélio Kistemann Junior – Universidade Federal de Juiz de Fora
Profª Drª Neiva Maria de Almeida – Universidade Federal da Paraíba
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Profª Drª Priscila Tessmer Scaglioni – Universidade Federal de Pelotas
Prof. Dr. Sidney Gonçalo de Lima – Universidade Federal do Piauí
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista

Diagramação: Daphynny Pamplona
Correção: Gabriel Motomu Teshima
Indexação: Amanda Kelly da Costa Veiga
Revisão: Os autores
Organizadora: Lilian Coelho de Freitas

Dados Internacionais de Catalogação na Publicação (CIP)

C691 Coleção desafios das engenharias: engenharia de computação 3 / Organizadora Lilian Coelho de Freitas. – Ponta Grossa - PR: Atena, 2021.

Formato: PDF

Requisitos de sistema: Adobe Acrobat Reader

Modo de acesso: World Wide Web

Inclui bibliografia

ISBN 978-65-5983-619-2

DOI: <https://doi.org/10.22533/at.ed.192212911>

1. Engenharia de computação. I. Freitas, Lilian Coelho de (Organizadora). II. Título.

CDD 621.39

Elaborado por Bibliotecária Janaina Ramos – CRB-8/9166

Atena Editora

Ponta Grossa – Paraná – Brasil

Telefone: +55 (42) 3323-5493

www.atenaeditora.com.br

contato@atenaeditora.com.br

DECLARAÇÃO DOS AUTORES

Os autores desta obra: 1. Atestam não possuir qualquer interesse comercial que constitua um conflito de interesses em relação ao artigo científico publicado; 2. Declaram que participaram ativamente da construção dos respectivos manuscritos, preferencialmente na: a) Concepção do estudo, e/ou aquisição de dados, e/ou análise e interpretação de dados; b) Elaboração do artigo ou revisão com vistas a tornar o material intelectualmente relevante; c) Aprovação final do manuscrito para submissão.; 3. Certificam que os artigos científicos publicados estão completamente isentos de dados e/ou resultados fraudulentos; 4. Confirmam a citação e a referência correta de todos os dados e de interpretações de dados de outras pesquisas; 5. Reconhecem terem informado todas as fontes de financiamento recebidas para a consecução da pesquisa; 6. Autorizam a edição da obra, que incluem os registros de ficha catalográfica, ISBN, DOI e demais indexadores, projeto visual e criação de capa, diagramação de miolo, assim como lançamento e divulgação da mesma conforme critérios da Atena Editora.

DECLARAÇÃO DA EDITORA

A Atena Editora declara, para os devidos fins de direito, que: 1. A presente publicação constitui apenas transferência temporária dos direitos autorais, direito sobre a publicação, inclusive não constitui responsabilidade solidária na criação dos manuscritos publicados, nos termos previstos na Lei sobre direitos autorais (Lei 9610/98), no art. 184 do Código Penal e no art. 927 do Código Civil; 2. Autoriza e incentiva os autores a assinarem contratos com repositórios institucionais, com fins exclusivos de divulgação da obra, desde que com o devido reconhecimento de autoria e edição e sem qualquer finalidade comercial; 3. Todos os e-book são *open access*, desta forma não os comercializa em seu site, sites parceiros, plataformas de *e-commerce*, ou qualquer outro meio virtual ou físico, portanto, está isenta de repasses de direitos autorais aos autores; 4. Todos os membros do conselho editorial são doutores e vinculados a instituições de ensino superior públicas, conforme recomendação da CAPES para obtenção do Qualis livro; 5. Não cede, comercializa ou autoriza a utilização dos nomes e e-mails dos autores, bem como nenhum outro dado dos mesmos, para qualquer finalidade que não o escopo da divulgação desta obra.

APRESENTAÇÃO

A Atena Editora tem a honra de presentear o público em geral com a série de *e-books* intitulada “*Coleção desafios das engenharias: Engenharia de computação*”. Em seu terceiro volume, esta obra tem o objetivo de divulgar aplicações tecnológicas da Engenharia de Computação na resolução de problemas atuais, com o intuito de facilitar a difusão do conhecimento científico produzido em várias instituições de ensino e pesquisa do país.

Organizado em 20 capítulos, este volume apresenta temas como utilização de aprendizagem de máquina na avaliação de riscos de infecção por COVID-19; dispositivos automatizados para administração de remédios; comunicação científica apoiada por realidade aumentada; métodos de elementos finitos aplicados na análise de materiais para indústria aeronáutica; aplicações de processamento digital de imagens e de algoritmos genéticos; entre diversas outras aplicações da automação e do desenvolvimento de *software*, combinados para melhorar as atividades do nosso dia-a-dia.

Dessa forma, esta obra contribuirá para aprimoramento do conhecimento de seus leitores e servirá de base referencial para futuras investigações.

Os organizadores da Atena Editora, agradecem especialmente os autores dos diversos capítulos apresentados, parabenizam a dedicação e esforço de cada um, os quais viabilizaram a construção deste trabalho.

Boa leitura.

Lilian Coelho de Freitas

SUMÁRIO

CAPÍTULO 1..... 1

EVALUATING THE RISK OF COVID-19 INFECTION BASED ON MACHINE LEARNING OF SYMPTOMS AND CONDITIONS VERSUS LABORATORY METHODS

Daniel Mário de Lima
João Henrique Gonçalves de Sá
Ramon Alfredo Moreno
Marina de Fátima de Sá Rebelo
José Eduardo Krieger
Marco Antonio Gutierrez

 <https://doi.org/10.22533/at.ed.1922129111>

CAPÍTULO 2..... 16

DISPOSITIVO AUTOMATIZADO PARA ADMINISTRAÇÃO DE REMÉDIOS

João Roberto Silva Teixeira
Alessandro Mainardi de Oliveira
Ricardo Neves de Carvalho

 <https://doi.org/10.22533/at.ed.1922129112>

CAPÍTULO 3..... 22

INTEGRAÇÃO ENTRE DADOS TEXTUAIS DE PRONTUÁRIOS ELETRÔNICOS DO PACIENTE (PEPS) E TERMINOLOGIAS CLÍNICAS

Amanda Damasceno de Souza
Eduardo Ribeiro Felipe
Fernanda Farinelli
Jeanne Louize Emygdio
Livia Marangon Duffles Teixeira
Maurício Barcellos Almeida

 <https://doi.org/10.22533/at.ed.1922129113>

CAPÍTULO 4..... 35

COMPARATIVE ANALYSIS OF THE PERFORMANCE OF A ENRICHED MIXED FINITE ELEMENT METHOD WITH STATIC CONDENSATION FOR POISSON PROBLEMS

Ricardo Javier Hanco Ancori
Jose Diego Ayñayanque Pastor
Rómulo Walter Condori Bustincio
Eliseo Daniel Velasquez Condori
Roger Edwar Mestas Chávez
Fermín Flavio Mamani Condori
Jorge Lizardo Díaz Calle

 <https://doi.org/10.22533/at.ed.1922129114>

CAPÍTULO 5..... 45

COMPORTAMENTO DE PAREDE DE ALVENARIA ESTRUTURAL EM SITUAÇÃO DE INCÊNDIO: ANÁLISE NUMÉRICA

Jean Marie Désir

Luana Zanin

 <https://doi.org/10.22533/at.ed.1922129115>

CAPÍTULO 6..... 58

COMUNICAÇÃO CIENTÍFICA APOIADA POR REALIDADE AUMENTADA: O CASO DO APLICATIVO AUMENTANDO KIRIMURÊ

Vinícius Pires de Oliveira
Fernanda Vitória Nascimento Lisboa
Jéssica Duarte Souza
Brisa Santana Brasileiro
Hilma Maria Passos de Oliveira
Ingrid Winkler
Andrea de Matos Machado
Karla Schuch Brunet

 <https://doi.org/10.22533/at.ed.1922129116>

CAPÍTULO 7..... 64

CONTEXTUALIZAÇÃO DO CPS DE UMA CÉLULA ROBÓTICA, ATRAVÉS DO GÊMEO DIGITAL UTILIZANDO PROTOCOLO DE COMUNICAÇÃO OPC UA

Rogério Adas Pereira Vitalli

 <https://doi.org/10.22533/at.ed.1922129117>

CAPÍTULO 8..... 75

DESENVOLVIMENTO DE UMA ARQUITETURA DE SOFTWARE BASEADA EM CENÁRIOS ARQUITETURAIS, MEMORANDOS TÉCNICOS E VISÕES DO MODELO 4+1

Everson Willian Pereira Bacelli
Bruno Ferreira Cardoso
Wilson Vendramel

 <https://doi.org/10.22533/at.ed.1922129118>

CAPÍTULO 9..... 90

DEVELOPMENT OF AN AIDING TOOL FOR THE OPTIMAL DETAIL OF ACTIVE REINFORCEMENT USING GENETIC ALGORITHM

Victória Carino Neves
Guilherme Coelho Gomes Barros

 <https://doi.org/10.22533/at.ed.1922129119>

CAPÍTULO 10..... 106

ANÁLISE DOS EFEITOS DA MÉTRICA DE DISTÂNCIA NA EXTRAÇÃO DE CONJUNTOS DE SIMILARIDADE

André Eduardo Alessi
Bruno Duarte
Ives Renê Venturini Pola
Dalcimar Casanova
Marco Antonio de Castro Barbosa

 <https://doi.org/10.22533/at.ed.19221291110>

CAPÍTULO 11	119
ESTUDO SOBRE AUTOMATIZAÇÃO DE EQUIVALÊNCIA DE FUNÇÕES	
Lucas Fernando Frighetto Fábio Hernandez	
 https://doi.org/10.22533/at.ed.19221291111	
CAPÍTULO 12	142
ESTUDO SOBRE O CONTROLE REMOTO DE DISPOSITIVOS MICROCONTROLADOS UTILIZANDO DISPOSITIVOS MÓVEIS	
João Vítor Fernandes Dias Fermín Alfredo Tang Montané	
 https://doi.org/10.22533/at.ed.19221291112	
CAPÍTULO 13	163
HERRAMIENTAS TECNOLÓGICAS APLICADAS EN EL DIBUJO ASISTIDO POR COMPUTADORA EN LA MODALIDAD A DISTANCIA	
Liliana Eneida Sánchez Platas Celia Bertha Reyes Espinoza Olivia Allende Hernández	
 https://doi.org/10.22533/at.ed.19221291113	
CAPÍTULO 14	174
HISTÓRICO DAS MULHERES NA TECNOLOGIA DA INFORMAÇÃO E ANÁLISE DA PARTICIPAÇÃO FEMININA NOS CURSOS SUPERIORES DO BRASIL	
Vívian Ludimila Aguiar Santos Thales Francisco Mota Carvalho Maria do Socorro Vieira Barreto	
 https://doi.org/10.22533/at.ed.19221291114	
CAPÍTULO 15	186
IDENTIFICAÇÃO DO MODELO DINÂMICO DE UMA TURBINA EÓLICA: ESTUDO DE CASO DA NORDTANK NTK 330F	
Gustavo Almeida Silveira de Souza Edgar Campus Furtado Leandro José Evilásio Campos Cristiane Medina Finzi Quintão	
 https://doi.org/10.22533/at.ed.19221291115	
CAPÍTULO 16	199
COMFORT IN VIBRATIONS FOR THE STEEL-CONCRETE COMPOSITE FLOORS: AN APPRAISAL FOR REVIEW OF ABNT NBR 8800:2008	
João Vitor V. Freire André V. Soares Gomes Adenílcia Fernanda G. Calenzani Johann A. Ferrareto	
 https://doi.org/10.22533/at.ed.19221291116	

CAPÍTULO 17	224
FINITE ELEMENT METHOD APPLIED TO MECHANICAL ANALYSIS OF AERONAUTICAL RIBS IN CARBON FIBER AND 7075 ALUMINUM ALLOY	
Alex Fernandes de Souza	
 https://doi.org/10.22533/at.ed.19221291117	
CAPÍTULO 18	236
MÉTODO PARA CALCULAR A ÁREA DE SUPERFICIAL DE RAÍZES POR PROCESSAMENTO DIGITAL DE IMAGENS	
Marcio Hosoya Name	
 https://doi.org/10.22533/at.ed.19221291118	
CAPÍTULO 19	244
LOCAL MESHFREE METHOD OPTIMIZATION WITH GENETICALGORITHMS	
Wilber Vélez	
Flávio Mendonça	
Artur Portela	
 https://doi.org/10.22533/at.ed.19221291119	
CAPÍTULO 20	258
NAVEGACIÓN VIRTUAL 2D Y 3D EN UN ENTORNO WEB	
Víctor Tomás Tomás Mariano	
Felipe de Jesús Núñez Cárdenas	
Jorge Hernández Camacho	
Isaura Argüelles Azuara	
Guillermo Canales Bautista	
 https://doi.org/10.22533/at.ed.19221291120	
SOBRE A ORGANIZADORA	268
ÍNDICE REMISSIVO	269

DESENVOLVIMENTO DE UMA ARQUITETURA DE SOFTWARE BASEADA EM CENÁRIOS ARQUITETURAIS, MEMORANDOS TÉCNICOS E VISÕES DO MODELO 4+1

Data de aceite: 01/11/2021

Data de submissão: 06/09/2021

Everson Willian Pereira Bacelli

Faculdade de Tecnologia da Zona Leste
São Paulo – São Paulo

Bruno Ferreira Cardoso

Faculdade de Tecnologia da Zona Leste
São Paulo – São Paulo

Wilson Vendramel

Faculdade de Tecnologia da Zona Leste
São Paulo – São Paulo
<http://lattes.cnpq.br/1507057519725073>

RESUMO: Inúmeras soluções nascem e rapidamente são substituídas por outras de igual funcionalidade por não se aterem aos atributos de qualidade que tem impacto sobre o negócio, isso denota que o hábito de ir direto para a implementação é um tanto quando descabido, pois, problemas que poderiam ser habilmente resolvidos no princípio dos projetos são propagados em retrabalhos, manutenções onerosas e sistemas inflexíveis ao longo do tempo. Isso poderia ser evitado com investimento na arquitetura de software, uma vez que concorre para antecipar a estrutura das soluções evitando inúmeros percalços quando são muito mais baratos de resolver e não chegam aos usuários. Diante do exposto, este trabalho tem o objetivo de desenvolver uma arquitetura de software para um E-commerce de venda de frutas, legumes

e verduras, seguindo como base as atividades de um ciclo de vida arquitetural. Para tal, este estudo adotou uma investigação experimental baseada em estudo de caso, explorando as seguintes atividades do ciclo de vida arquitetural: a) definição dos casos de uso principais do modelo de negócio; b) análise dos requisitos arquiteturalmente significativos a partir de cenários arquiteturais e memorandos técnicos; c) criação e documentação da arquitetura com o modelo de visões 4+1; d) implementação da arquitetura com base nos estilos arquiteturais SOA e EDA e; e) testes arquiteturais para validar os atributos de qualidade. Como resultados obtidos, destacam-se a aderência entre a solução de software e a arquitetura concebida em relação às prescrições de design, as evidências de que essa arquitetura suporta os cenários arquiteturais mediante os testes executados e a própria documentação da arquitetura de software em múltiplas visões.

PALAVRA-CHAVE: Arquitetura de Software, Ciclo de Vida Arquitetural, Cenários Arquiteturais, Memorandos Técnicos, Visões do Modelo 4+1.

DEVELOPMENT OF A SOFTWARE ARCHITECTURE BASED ON ARCHITECTURAL SCENARIOS, TECHNICAL MEMOS AND 4+1 VIEW MODEL

ABSTRACT: Numerous solutions are born and are quickly replaced by others with equal functionality because they do not adhere to the quality attributes that have an impact on the business, this denotes that the habit of going straight to implementation is somewhat unreasonable, as problems that could be skillfully resolved early in the project

are propagated into rework, have costly maintenance, and inflexible systems over time. This could be avoided with investment in software architecture, since it contributes to anticipate the structure of solutions, avoiding numerous mishaps when they are much cheaper to solve and do not reach users. Given the above, this work aims to develop a software architecture for an E-commerce for the sale of fruits and vegetables, following as a basis the activities of an architectural life cycle. To this end, this study adopted an experimental investigation based on a case study, exploring the following activities of the architectural lifecycle: a) definition of the main use cases of the business model; b) analysis of architecturally significant requirements from architectural scenarios and technical memos; c) creation and documentation of the architecture with the 4+1 view model; d) implementation of the architecture based on SOA and EDA architectural styles and; e) architectural tests to validate quality attributes. As results, the adherence between the software solution and the architecture conceived in relation to the design prescriptions, the evidence that this architecture supports the architectural scenarios through the executed tests and the documentation of the software architecture in multiple views.

KEYWORDS: Software Architecture, Architecture Lifecycle, Architectural Scenarios, Technical Memos, 4+1 View Model.

1 | INTRODUÇÃO

Arquitetos de software dispõe de pouco ou nenhum apoio frente ao desafio de conceber arquiteturas de software, como resultado enfrentam dificuldades ao definir o *design* de soluções que atendam as metas de negócio e os atributos de qualidade (RODRIGUES; PACE; SORIA, 2018). Essa visão é reforçada ao se tomar ciência de que há uma carência de métodos sistemáticos para concepção de arquiteturas (ZHAO *et al.*, 2018).

Martin (2019) concorda com esse entendimento quando pontua que é comum os programadores, frente a novos desafios, 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, conseqüentemente isso resulta em sistemas rígidos, caros e difíceis de manter, comprometendo assim sua longevidade.

“A arquitetura de software de um sistema é o conjunto de estruturas necessárias para raciocinar sobre ele, que compreende elementos de software, suas relações, e propriedades de ambos.” (BASS; CLEMENTS; KAZMAN, 2012, p. 4, tradução nossa). Nesse sentido, a arquitetura é útil, uma vez que contribui para se ter uma visão geral da solução antes dessa existir, o que facilita a análise do projeto frente aos requisitos, simplifica reflexões sobre alternativas e põe em questão mudanças quando é relativamente fáceis de serem realizadas, minimizando os riscos do projeto (PRESMAN, 2011). Além disso, é importante para o cumprimento dos requisitos, tendo em vista que exercem influência dominante sobre os atributos de qualidade, os quais têm forte impacto sobre o sistema ou certas funcionalidades, e por isso podem inviabilizar soluções (SOMMERVILLE, 2011).

Por isso, o objetivo deste trabalho consiste em propor um conjunto de técnicas,

modelos e ferramentas que estruturam o ciclo de vida arquitetural como estratégia para guiar os programadores no esforço de desenvolvimento de software com ênfase na arquitetura.

Para tanto, o trabalho apresenta conceitos importantes sobre arquitetura de software e ciclo de vida arquitetural de Bass, Clements e Kazman (2012) e o explora com o apoio de algumas ferramentas obtidas na literatura consultada: primeiro, os requisitos arquiteturais são extraídos dos cenários mais importantes da solução; em seguida, são classificados e analisados com FURPS+ como sugerido por Eeles (2001) e uma adaptação dos memorandos técnicos de Larman (2007); depois disso, a arquitetura é elaborada em múltiplas visões a partir do modelo 4+1 de Kruchten (1995) e implementada de acordo com as prescrições de *design*; por fim, testes arquiteturais são apresentados para validar a capacidade da solução de atender os requisitos arquiteturais e algumas considerações finais são expostas.

2 | CICLO DE VIDA ARQUITETURAL

O ciclo de vida arquitetural proposto por Bass, Clements e Kazman (2012) consiste em uma estratégia que contribui para o desenvolvimento de soluções com foco na arquitetura. Esse ciclo é composto por um conjunto de sete atividades que expressam um modelo de criação, *design*, implementação e evolução de uma arquitetura de software para um dado sistema ou aplicação-alvo. A atividade 1 propõe o desenvolvimento de um caso de uso de negócio que coloca em questão a finalidade do projeto, os benefícios que dele se espera e serve de guia para cursos de ação; a atividade 2 visa definir e compreender os requisitos arquiteturalmente significativos, uma vez que sua presença ou ausência tem impacto substancial sobre o sistema atender ou não as necessidades dos stakeholders e sobre a configuração da arquitetura; a atividade 3 consiste na criação ou seleção de uma arquitetura e tem como ponto crítico a busca pela integridade conceitual, fruto de uma convergência entre requisitos importantes e decisões de *design*; a atividade 4, por sua vez, tem por objetivo documentar e comunicar a arquitetura para que opere como espinha dorsal do projeto, contribuindo para uma compreensão da estrutura da solução e suas implicações de forma inequívoca para os envolvidos; a atividade 5 busca analisar e avaliar a arquitetura e, para tanto, reflete sobre opções concorrentes frente aos atributos de qualidade; a atividade 6 tem como propósito implementar a solução em conformidade com a arquitetura, transformando *design* em software executável; por fim, a atividade 7 coloca em questão se o software resultante atende aos requisitos desejados.

3 | CICLO DE VIDA: DEFINIÇÃO DOS CASOS DE USO DA SOLUÇÃO

A atividade 1 do ciclo de vida consiste na definição de casos de uso de negócio para solução, isso é importante porque “As boas arquiteturas devem ser centradas em casos de uso para que os arquitetos possam descrever com segurança as estruturas que suportam esses casos de uso, sem se comprometerem com *frameworks*, ferramentas e ambientes.”

(MARTIN, 2019, p. 153).

Neste estudo, utilizou como ponto de partida para explorar o ciclo de vida arquitetural, os modelos, ferramentas e o desenvolvimento dos cenários principais de uma solução Web para apoiar feiras-livres a ingressarem no ambiente de vendas online, estes cenários são: compra, pagamento e entrega (CARDOSO; BACELLI, 2021).

3.1 Caso de Uso – Realização da compra

Ele tem início com o cliente autenticado no sistema fazendo a inclusão de itens nas quantidades desejadas no carrinho; finalizada essa etapa, esse é redirecionado para uma interface que descreve os itens e os valores e solicita confirmação. Ao proceder, o sistema solicita os dados de pagamentos, então, ele executa essa tarefa e recebe uma confirmação de inclusão do pedido com sucesso.

3.2 Caso de Uso – Realização do pagamento

Após a inclusão, o serviço de pagamento da solução é responsável por realizar as operações para encaminhar os dados de pagamento para um serviço externo, mediante a confirmação dele, o cliente é notificado e os parceiros de negócio avisados de que há um pedido a ser entregue com todos os detalhes.

3.3 Caso de Uso – Realização da entrega

Após a confirmação do pagamento, o sistema atribui o pedido para um entregador e o notifica, juntamente com o feirante para que monte o pedido. E, por fim, ao realizar a entrega, o entregador a confirma no sistema e processo é finalizado.

4 | CICLO DE VIDA: COMPREENDER E ANALISAR A ARQUITETURA

As etapas 2 e 5 do ciclo de vida arquitetural colocam em perspectivas as atividades de compreender e analisar os requisitos arquiteturalmente significativos, respectivamente. Afinal a presença ou ausência destes atributos de qualidade importantes tendem a modificar profundamente a configuração da arquitetura e, portanto, a capacidade da solução de atender as necessidades dos stakeholders (BASS; CLEMENTS; KAZMAN, 2012).

Para apoiar esse desafio alguns requisitos importantes foram identificados e classificados utilizando FURPS+ como sugerido por Eeles (2001), e como resultado desse esforço foram identificados múltiplos requisitos arquiteturais: três de natureza funcional, dois classificados como confiabilidade, um com suportabilidade e, por fim, um plus (+) conforme observado na figura 1.

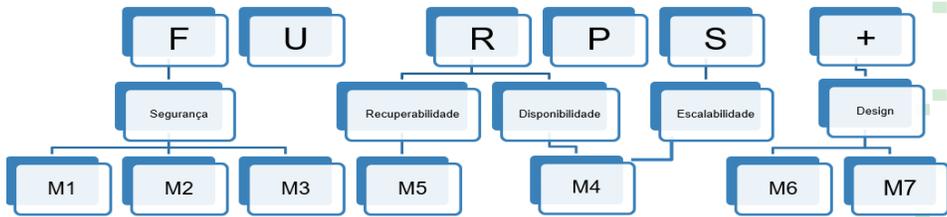


Figura 1 – FURPS+ e Memos Técnicos.

Fonte: Adaptado de Cardoso e Bacelli (2021).

Em seguida, esses atributos de qualidade foram submetidos a um detalhamento com apoio de uma adaptação dos memorandos técnicos proposto por Larman (2007) a fim de verificar sua relevância arquitetural. Isso culminou em 7 quadros que permitiram identificar com clareza o atributo em questão, determinar uma forma de perceber sua presença ou ausência na arquitetura, justificar sua importância e classificação como requisito arquiteturalmente significativo, determinar a solução técnica empregada e expõe alternativas consideradas.

Fator: Proteger os recursos expostos externamente da aplicação de acessos indevidos.
Motivação: A utilização de um token de autorização contribui diretamente para simplificar o gerenciamento de acesso aos recursos do sistema, esse benefício é obtido pela eliminação da necessidade de controle de vários estados que o usuário pode possuir na aplicação, que é constituída por diversos serviços. Ao mesmo tempo em que concorre para aumentar a segurança da aplicação pois evita a necessidade de trânsito de informações de acesso entre às requisições, e permuta uma validade temporária e parametrizável, reduzindo expressivamente a chance de elas serem interceptadas por agentes maliciosos.
Solução: Obter token de acesso utilizando protocolo oauth2 por meio de um serviço de autorização centralizado que gerenciará o acesso aos recursos protegidos da arquitetura
Alternativa: Permitir acesso aos recursos por meio do login e senha do usuário, estratégia desconsiderada pois exigiria o tráfego das suas credenciais em plain/text.

Quadro 1 – Memorando Técnico 1 – M1 (Funcional/Segurança).

Fonte: Adaptado de Cardoso e Bacelli (2021).

Fator: Proteção dos dados sensíveis de pagamento.
Motivação: Evitar a exposição de dados de pagamento dos clientes a pessoas indesejadas e todas as responsabilidades legais decorrentes desse problema.
Solução: Utilização de técnica de criptografia com chave assimétrica (pública e privada) para proteção dos dados sensíveis de pagamento.
Alternativa: Nenhuma alternativa.

Quadro 2 – Memorando Técnico 2 – M2 (Funcional/Segurança).

Fonte: Adaptado de Cardoso e Bacelli (2021).

Fator: Validação dos dados do pedido.
Motivação: O processamento de requisições fraudulentas pode ter como consequência prejuízos para o negócio com recursos gastos indevidamente; associado a perda de credibilidade junto a clientes e parceiros de negócios; e gerar cargas de trabalho decorrentes de esforços de auditoria.
Solução: Implementação de um algoritmo de validação para verificar o preenchimento dos dados de pagamento, bem como a validade dos valores relacionados ao pedido, para isso os dados relacionados os produtos serão comparados com os persistidos em disco (Banco Dados).
Alternativa: Delegar a validação dos dados de pagamento apenas para as APIs externas.

Quadro 3 – Memorando Técnico 3 - M3 (Funcional/Segurança).

Fonte: Adaptado de Cardoso e Bacelli (2021).

Fator: Serviços de compra e entrega escaláveis e com alta disponibilidade.
Motivação: Essa estratégia se mostra importante em três perspectivas. A primeira está relacionada com a experiência oferecida ao cliente que se vendo frustrado no seu objetivo de realizar pedidos, pode atribuir uma imagem negativa ao negócio e não realizar novas compras no sistema. Segundo, está associado com o suporte dado para operação, isto é, feirantes precisam dimensionar compras e precisam de informações em tempo hábil para se planejar e entregadores precisam obter informações de localização e volume para realizar as entregas de forma precisa. Portanto, a indisponibilidade ou falhas podem comprometer objetivos importantes do negócio. Por fim, a escalabilidade por replicação de nós de processamento foi adotada, pois a aquisição de máquinas físicas ou serviços de cloud com mais memória e número de CPUs é muito mais custosa financeiramente que a aquisição de múltiplas máquinas de poder computacional mais modesto, além disso a distribuição dos serviços em múltiplos nós de processamento evita um ponto único de falha.
Solução: Usar uma estratégia de replicação dos serviços de compra e entrega do sistema em nós de processamento distintos para contornar possíveis problemas de indisponibilidade.
Alternativa: Utilização de técnica de escalabilidade vertical, adicionando recursos computacionais nas máquinas dos serviços de compra e entrega. Desconsiderado por questões de custo e ponto único de falha.

Quadro 4 – Memorando Técnico 4 – M4 (Confiabilidade/Disponibilidade e Suportabilidade/ Escalabilidade).

Fonte: Adaptado de Cardoso e Bacelli (2021).

Fator: Recuperabilidade do sistema frente a problemas no sistema de conexão com o sistema externo de pagamento.
Motivação: O pagamento é uma função muito importante do sistema pois representa entrada de receita para o negócio, confirmação de pagamento para cliente, mobilização de estoque
Solução: Adotar um Serviço de pagamento interno responsável por coordenar as operações de submissão de pagamento para sistemas externos, evitando que falhas neles prejudiquem o processamento dos pagamentos e o desempenho do sistema. Para tanto, esse Serviço contará com uma fila de mensageria que será responsável por persistir os dados de pagamentos em disco, evitando que estas informações sejam perdidas em caso de falha no sistema de pagamento externo ou haja uma carga de trabalho superior a capacidade do sistema. Também foi incluso um limite de 3 vezes para nova inclusão de um pagamento, a fim de evitar que as requisições persistam por tempo indeterminado no serviço.
Alternativa: Realizar a persistência dos dados sem a fila de mensageria, mas essa opção foi desconsiderada pois aumentava o acoplamento entre os serviços da arquitetura uma vez que precisariam se conhecer.

Quadro 5 - Memorando Técnico 5 – M5 (Confiabilidade/Recuperabilidade).

Fonte: Adaptado de Cardoso e Bacelli (2021).

Fator: Aplicação organizada em serviços independentes com responsabilidades funcionais distintas. Serviços internos da arquitetura se comunicam através de eventos.
Motivação: O uso de SOA concorre para aumentar a capacidade de adaptação da solução a ao ambiente de negócio, favorece o reuso de componentes e minimiza o esforço, o tempo e o custo de desenvolvimento; além disso, concorre para tornar os componentes da arquitetura independentes e por isso mais resilientes a problemas. Por outro lado, EDA contribui para que a arquitetura responda de forma mais inteligente a estímulos importantes do sistema, garantindo que esses sejam tolerantes a eventuais problemas de comunicação promovendo maior confiabilidade no que tange aos dados de negócio importantes que estão transitando entre os serviços. Além disso, a utilização das filas de mensagens baseadas no modelo Orientado a Eventos reduz o acoplamento que seria causado por um modelo puramente SOA.
Solução: Organizar a arquitetura da solução a partir do estilo baseado em serviços a fim de definir claramente as responsabilidades, tornando as unidades lógicas da aplicação independentes em termos de recursos e propósito. Além disso, também será implementado uma abordagem baseada em eventos para orquestrar a comunicação entre os serviços internos de forma pouco acoplada, tolerante a falhas e assíncrona.
Alternativa: Elaborar a arquitetura baseada em micro serviços, alternativa abandonada em razão do tempo do projeto e a complexidade imposta por essa alternativa.

Quadro 6 – Memorando Técnico 6 – M6 (Design).

Fonte: Adaptado de Cardoso e Bacelli (2021).

Fator: Serviços da aplicação com endereço físico revelados dinamicamente
Motivação: A utilização de um descobridor de serviços traz a arquitetura maior capacidade de escalar horizontalmente, facilita o processo de novos nós de processamento se cadastrarem como disponíveis, evitando assim a necessidade dos consumidores desses nós conhecerem em tempo de compilação seus endereços físicos.
Solução: Implementação do componente discovery do Spring Cloud Netflix que permite o registro dos serviços da arquitetura em tempo de execução, além de ser mantido com atualizações periódicas.
Alternativa: Centralizar os endereços conhecidos dentro do balanceador de cargas, isso limitaria a capacidade da aplicação escalar uma vez que seria necessário atualizar os endereços físicos manualmente a cada inclusão de nós de processamento.

Quadro 7 – Memorando Técnico 7 – M7 (Design e Suportabilidade/Escalabilidade).

Fonte: Adaptado de Cardoso e Bacelli (2021).

Dessa forma, os memorandos técnicos são uma estratégia que contribuíram para ampliar o entendimento sobre os atributos de qualidades identificados, justificando sua caracterização como requisitos arquiteturais. Ao mesmo tempo que abriram espaço para explicar as soluções selecionadas e alternativas consideradas. Portanto, definindo alvos a serem perseguidos na elaboração das visões arquiteturais.

5 | CICLO DE VIDA: CRIAR E DOCUMENTAR A ARQUITETURA

As atividades 3 e 4 do ciclo estão relacionadas a criar ou selecionar uma e documentar e comunicar uma arquitetura, respectivamente. Foram organizadas a partir do modelo 4+1 de Kruchten (1995), o qual propõe o desenvolvimento da arquitetural em múltiplas visões como estratégia para lidar com a complexidade imposta pelos cenários e seus atributos

de qualidade. A figura 2 apresenta uma relação entre os memorandos técnicos de Larman (2007) descritos na seção anterior com as 4 visões. Essa figura mostra que as visões são complementares e contribuem para uma melhor compreensão da relação entre solução, seus atributos de qualidade e estruturas arquiteturais.

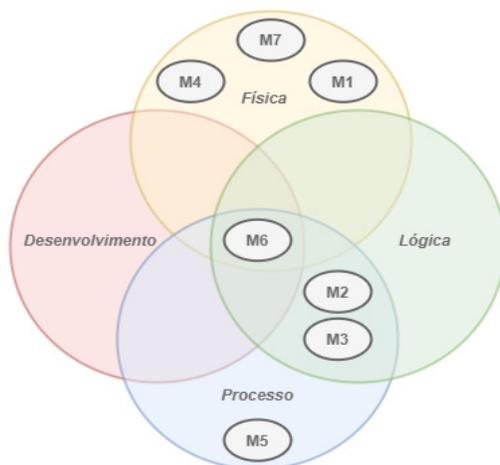


Figura 2 – Memorandos e Visões.

Fonte: Adaptado de Cardoso e Bacelli (2021).

5.1 Visão Lógica

A visão lógica oferece suporte especial aos requisitos funcionais, permitindo observar as responsabilidades e características das classes e objetos do sistema com base em abstrações retiradas do domínio do negócio KRUCHTEN (1995). No estudo foi elaborado a partir do diagrama de classes que apresenta uma perspectiva estrutural das soluções. A figura 3 apresenta parcialmente uma visão relativa a M2, destacando o padrão de projeto *Chain of Responsibility* como solução plugável para o tratamento do objeto de pagamento nos cenários de compra (CARDOSO; BACELLI, 2021).

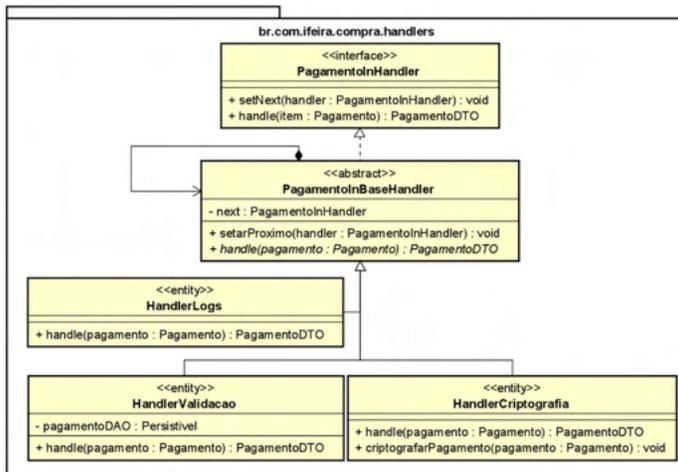


Figura 3 – Visão Lógica - Chain of Responsibility.

Fonte: Adaptado de Cardoso e Bacelli (2021).

5.2 Visão de Desenvolvimento

A arquitetura de desenvolvimento é organizada ao redor dos módulos, bibliotecas ou serviços, elementos que servem de base para atribuição de responsabilidade as equipes de desenvolvimento. Em geral, essa visão apresenta uma estrutura em camadas com ênfase nas relações de importação e exportação dos módulos e serviços, encapsulamento do funcionamento e suas interfaces KRUCHTEN (1995). A figura 3 apresenta a visão de desenvolvimento elaborada para a aplicação (CARDOSO; BACELLI, 2021). Essa figura enfatiza as dependências estruturais entre os três serviços que compõem a arquitetura, reforçando a presença de M6 e também revela as dependências que possuem entre si.

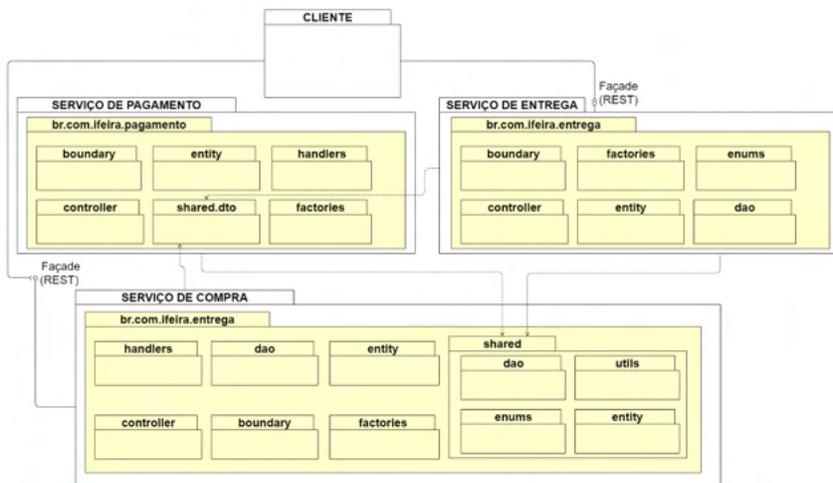


Figura 4 – Visão de Desenvolvimento.

Fonte: Adaptado de Cardoso e Bacelli (2021).

5.3 Visão de Processo

A visão de processo coloca em questão o funcionamento do sistema em tempo de execução e o consumo de recursos dele decorrente, contribuindo para destacar aspectos de qualidade como desempenho, disponibilidade, distribuição, simultaneidade, integridade e tolerância a falhas, inclusive a arquitetura de processo pode ser elaborada em diferentes perspectivas para colocar em destaque aspectos de interesse, sendo uma visão de mais alto nível KRUCHTEN (1995).

A figura 4 apresenta uma visão macro de processo e contribui para complementar o entendimento de M6, pois evidencia a comunicação entre os serviços via fila de mensageria com base em eventos, demonstrando a presença de EDA (Eventi Driven Architecture) e SOA (Service Oriented Architecture) na arquitetura. Essa figura também permite o entendimento de que M5 trata da persistência dos dados de pagamento em disco, na hipótese de o sistema externo de pagamento se apresentar inoperante ou inalcançável durante o envio de requisições (CARDOSO; BACELLI, 2021).

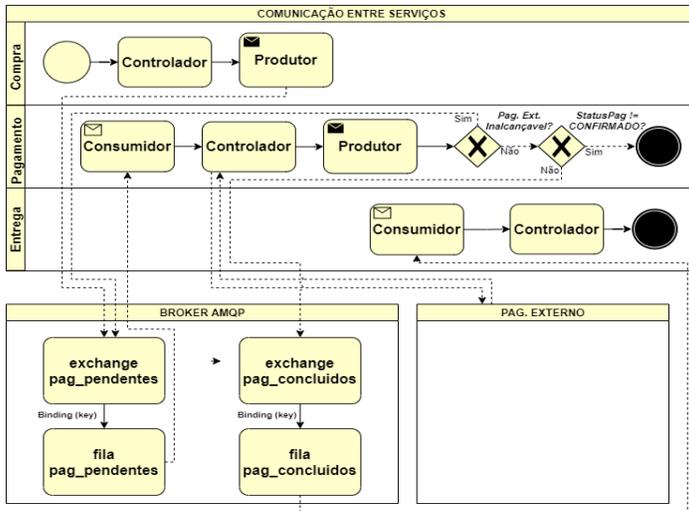


Figura 5 – Visão de Processo.

Fonte: Cardoso e Bacelli (2021).

5.4 Visão Física

A arquitetura física tem por objetivo ilustrar em quais componentes físicos e/ou nós de processamento o software irá funcionar, isto é, redes, processos, tarefas e objetos são mapeados para o hardware. É importante que seja altamente flexível para que os diferentes ambientes tenham o mínimo de impacto sobre o código-fonte. Essa perspectiva contribui diretamente para colocar em relevo aspectos de qualidade do software como disponibilidade, confiabilidade, desempenho e escalabilidade KRUCHTEN (1995).

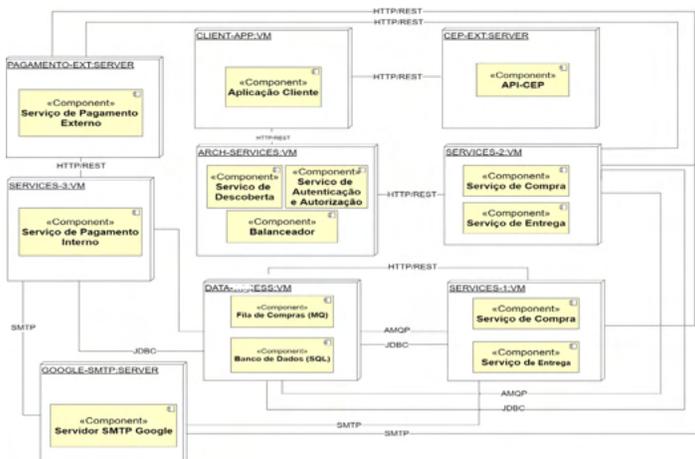


Figura 6 – Visão Física
Fonte: Cardoso e Bacelli (2021).

A representação em questão ilustra que a aplicação é constituída de nós de processamento independentes entre si em termos de infraestrutura como memória, processamento e canais de conexão. Isso está em acordo com o M6 que estabelece que a solução deve estar organizada em serviços com baixo grau de acoplamento, conforme o estilo SOA. Também se pode observar os serviços de compra e entrega replicados, estratégia conhecida como escalabilidade horizontal conforme Pressman (2011) que concorre para aumentar a capacidade do sistema em atender um maior número de requisições, assim como diminuir sua sensibilidade frente a falhas, aspectos presentes em M4.

6 | CICLO DE VIDA: IMPLEMENTAR A ARQUITETURA

A implementação consiste na penúltima atividade do ciclo de vida arquitetural e é caracterizada pelo aumento do número de detalhes e conseqüentemente da complexidade, é responsável por converter a arquitetura em software executável. Não por menos, Sommerville (2011) defende que se trata de uma atividade crítica, haja vista que uma série de decisões são postas em prática, como escolhas de tecnologias, linguagens, reuso, frameworks etc.

Nesse estudo optou-se pela linguagem “Java” por ser amplamente disseminada e dispor do framework Spring Boot, que facilitou a implementação das necessidades prescritas no *design* concebido:

Spring Cloud Security: contribuiu para a implementação do protocolo Auth2, responsável pela autorização de acesso a recursos expostos;

Spring Cloud Netflix: permitiu com facilidade pôr em prática o balanceador de carga e o descobridor de serviços.

Além disso, para lidar com o *build* e dependências da aplicação, optou-se pelo Gradle; também foi selecionada a fila de mensageria AMQP para minimizar o acoplamento decorrente da comunicação entre os serviços, o banco de dados MYSQL para persistência e o ambiente *cloud* Azure para hospedar a aplicação.

Independentemente das tecnologias utilizadas, Bass, Clements e Kazman (2012) ressaltam que as decisões de *design* devem ser entendidas como prescrições aos programadores, pois não adianta se dedicar tanto tempo e esforço para se definir uma arquitetura que habilita certos atributos de qualidade, se ela não for respeitada.

7 | CICLO DE VIDA: TESTES ARQUITETURAIIS

Os testes são a última fase do ciclo de vida arquitetural e visam ilustrar se a arquitetura implementada suporta os requisitos arquiteturalmente significativos. Isso está de acordo com Martin (2019, p. 408) quando comenta que “Há tempos se sabe que a testabilidade é um atributo das boas arquiteturas.” Por isso, no contexto desse estudo dez testes foram realizados para evidenciar que a arquitetura suporta os requisitos arquiteturais, o objetivo de cada testes está descrito no quadro 8.

Teste 1 - Obter acesso a um recurso protegido;
Descrição: Envio 100 requisições com token de autorização válido;
Teste 2 - Negar acesso a um recurso protegido
Descrição: Envio de 100 requisições com token de autorização inválido;
Teste 3 - Inclusão dos dados na fila de mensageria de forma protegida
Descrição: Enviado 100 inclusões de pedidos para fila de mensageria com serviço de pagamento inoperante
Teste 4 - Bloquear inclusão de pedidos por dados adulterados
Descrição: Enviado 100 requisições para o serviço de pagamentos com dados internos da requisição modificados.
Teste 5 - Bloquear a inclusão de pedidos por dados de pagamento incompletos
Descrição: Enviada 100 requisições de inclusão com dados de pagamento incompletos
Teste 6 - Atender múltiplas requisições de compra
Descrição: Enviada 1.200 requisições para o serviço de pedido
Teste 7 - Lidar com indisponibilidade de nós de processamento
Descrição: Tornando nó de processamento indisponível em um cenário de 100 requisições simultâneas
Teste 8 - Evitar perda de dados por problemas externos
Descrição: Enviada 100 requisições para o sistema de pagamento, após tornar o sistema de pagamento externo indisponível.
Teste 9 - Demonstrar a independência dos serviços da aplicação
Descrição: Enviada 100 requisições para o sistema de pagamento externo, com os demais serviços desligados. Em seguida, o serviço de pagamento foi acionado e depois o de entrega.
Teste 10 - Inclusão de novos nós de processamento na arquitetura
Descrição: Realizada a inclusão de 2 novos nós de processamento da arquitetura.

Quadro 8 – Objetivo e Descrição dos Testes.

Fonte: Adaptado de Cardoso e Bacelli (2021).

Estes dez testes foram colocados em prática com o apoio da ferramenta Apache JMeter e os resultados são exibidos na figura 7. Nessa figura, é possível observar a linha azul representando as metas estabelecidas para cada teste e as colunas em laranja mostrando o resultado obtido com cada um. Também se observa em forma de comentário a qual memorando técnico, o teste está relacionado.

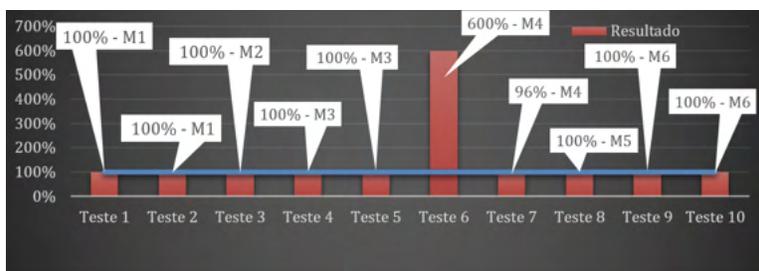


Figura 7 – Resultado dos Testes.

Fonte: Adaptado de Cardoso e Bacelli (2021).

Dessa forma, a partir dos dez testes realizados, é possível verificar que a arquitetura implementada apresenta as evidências previstas nos memorandos técnicos, suportando os

requisitos arquiteturais identificados e assegurando sua conformidade às características da arquitetura proposta.

8 | CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo apresentar um conjunto de técnicas, modelos e ferramentas ao redor do ciclo de vida arquitetural como estratégia para lidar com a lacuna de conhecimento quanto ao esforço de criar arquiteturas coerentes com objetivos de negócios e requisitos arquiteturais. Frente a essa problemática o estudo atingiu seu objetivo porque, primeiro, a seção 4 apresentou uma forma de descrever cenários de uso principais de soluções; que em seguida são analisados e organizados com os memorandos técnicos e o modelo FURPS+ na seção 5.

Na seção 6, é possível observar a arquitetura desenvolvida nas perspectivas lógica, de processo, física e de desenvolvimento com base no modelo 4+1 e isso permitiu observar as estruturas planejadas, suas relações e propriedades, evidenciando a presença dos requisitos arquiteturalmente relevantes no desenho da solução.

Além disso, observa-se nas seções 6 e 7, uma implementação aderente à arquitetura proposta, preservando as intenções e os requisitos arquiteturais almejados pelos autores quando dimensionaram a estrutura e os relacionamentos que compõem a solução, assim como, expõe uma sequência de testes arquiteturais que evidenciam que o *design* proposto suporta os requisitos arquiteturais.

Portanto, o trabalho em questão atingiu seu objetivo uma vez que apresenta um roteiro estruturado para programadores enfrentarem o desafio de desenvolver soluções sem renunciar a uma arquitetura alinhada com as necessidades do projeto. O estudo também abre espaço para futuros trabalhos colocarem a prova o guia em outros contextos, reforçando e refutando sua eficácia, também dá margem para proposição de novas ferramentas ou o refinamento das atuais.

REFERÊNCIAS

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. **Software Architecture in Practice**. 3.ed. New Jersey: Person Education, 2012.

CARDOSO, Bruno Ferreira; BACELLI, Everson Willian Pereira. **Desenvolvimento de uma Arquitetura Heterogênea Baseada em Cenários Arquiteturais, Memorandos Técnicos e Visões do Modelo 4+1**. 2021. 138f. Trabalho de Conclusão de Curso – Faculdade de Tecnologia da Zona Leste, São Paulo, 2021.

EELES, Peter. Capturing architectural requirements. **IBM Rational Developer Works**, 2001. Disponível em: https://www.researchgate.net/publication/329760910_Capturing_Architectural_Requirements. Consultado em: 8 dez 2020.

KRUCHTEN, Philippe. Architectural Blueprints - The “4+1” View Model of Software Architecture. **Institute of Electrical and Electronics Engineers Computer (IEEE) Society**, Califórnia, vol.12, n.6, p.42-50, Nov, 1995. DOI 10.1109/52.965801.

LARMAN, Craig. **Utilizando UML e Padrões**: Uma Introdução à Análise e ao Projeto Orientado a Objetos. 3.ed. Porto Alegre: Bookman, 2007.

MARTIN, Robert. **Arquitetura Limpa**: O Guia do Artesão para Estrutura e Design de Software. Rio de Janeiro: AltaBooks, 2019.

PRESSMAN, Roger. **Engenharia de Software**: Uma Abordagem Profissional. 7.ed. Porto Alegre: AMGH, 2011.

RODRÍGUEZ, Guillermo; DÍAZ-PACE, J. Andrés; SORIA, Álvaro. A Case-Based Reasoning Approach to Reuse Quality-Driven Designs in Service-Oriented Architectures. **Information Systems**, v. 77, p. 167-189, 2018.

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

ZHAO, Heng et al. SOA Patterns Selection and Application Based on Software Quality Requirements. In: **2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)**. IEEE, 2018. p. 361-364.

ÍNDICE REMISSIVO

A

Acoplamento termomecânico 44, 48, 52

Algoritmo genético (AG) 244

Alvenaria estrutural 4, 44, 48

Análise de imagem 235, 240, 241

Aprendizado de máquina 2

Arduino 17, 18, 19, 20, 141, 142, 144, 145, 146, 147, 148, 152, 154, 157, 158, 159, 160, 161

Arquitetura de software 5, 74, 75, 76

B

Balanced spaces 34

Biblioteconomia clínica 21

Bluetooth 141, 142, 143, 144, 146, 147, 148, 151, 152, 154, 155, 156, 157, 158, 159, 160, 177

C

Cenários arquiteturais 5, 74, 87

Ciclo de vida arquitetural 74, 76, 77, 85, 87

Comunicação científica 3, 5, 57, 58

Conjuntos de similaridade 5, 105, 107, 108, 116

Correlação 235, 236, 240

D

Dados complexos 105, 106, 107, 108

Design science research 57, 58, 59, 62

Desigualdade de gênero na TI 173, 174

Dibujo asistido por computadora 6, 162, 163, 164, 171

E

Educación a distancia 162, 164, 165, 168, 170, 171

Elementos finitos 3, 48, 52, 53, 223

Energia renovável 185

Equivalência de funções 6, 118

F

Fibra de carbono 223

G

Gêmeo digital 5, 63, 64, 68, 71

Grafos 105, 112, 259, 261

H

Herramientas tecnológicas 6, 162, 163, 164, 170

Histórico feminino na TI 173, 174

Human comfort 198

I

Identificação de sistemas 185, 188, 189

Idosos 16, 17, 20

Indústria 4.0 63, 65, 66, 67

Infecções por Coronavirus 2

Interoperabilidade 21, 23, 24, 25, 26, 30, 32, 63, 64, 66, 67

J

JavaCV 235, 236, 237, 240, 241

JavaScript 141, 142, 153, 263

L

Ligas de alumínio 223

M

Memorandos técnicos 5, 74, 76, 78, 80, 81, 86, 87

Método sem malha local 243, 244

Método sem malha local com integração reduzida (ILMF) 244

Métrica de distância 5, 105, 113, 116

Microcontrolador 17, 141, 152

Mixed finite elements 34

Mulheres na TI 173, 174, 182, 183

Mulheres nos cursos superiores de TI 173, 174

O

Ontologias 21, 22, 23, 24, 25, 29, 30, 31, 32

opencv 241

OpenCV 235, 236, 237, 240, 241

Optimal detailing 89

P

Poisson's equation 34, 36

Prestressed concrete 89, 90, 91, 92, 96, 103

R

Rami 4.0 65

RAMI 4.0 63, 64, 65, 66, 67, 68, 69, 71

Realidade aumentada 3, 5, 57, 58, 60, 62

Remédios 3, 4, 16, 17, 20

Resistência ao fogo 44, 45, 49, 50, 56

Resistência mecânica 50, 55, 223

Robotista 63

S

Sistemas ciberfísicos (CPS) 63, 64, 71

Static condensation 4, 34, 35, 36

Steel-concrete 6, 198, 199, 200, 202, 204, 205, 206, 216, 218, 221

T

Terminologias clínicas 4, 21, 23, 24, 25, 30

Teste de hipótese 105

U

Usinas eólicas 185

V

Vibrations 6, 198, 199, 212, 219, 220, 222

Visões do modelo 4+1 5, 74, 87

Visualização de dados 57

W

Wi-Fi 141, 142, 147, 148, 152, 153, 157, 158

COLEÇÃO

DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO 3

-  www.atenaeditora.com.br
-  contato@atenaeditora.com.br
-  [@atenaeditora](https://www.instagram.com/atenaeditora)
-  www.facebook.com/atenaeditora.com.br

COLEÇÃO

DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO 3

-  www.atenaeditora.com.br
-  contato@atenaeditora.com.br
-  [@atenaeditora](https://www.instagram.com/atenaeditora)
-  www.facebook.com/atenaeditora.com.br