

# Sistemas de Informação e Aplicações Computacionais

**Ernane Rosa Martins**  
(Organizador)

 **Atena**  
Editora  
Ano 2020

# Sistemas de Informação e Aplicações Computacionais

**Ernane Rosa Martins**  
(Organizador)

**Atena**  
Editora  
Ano 2020

**Editora Chefe**

Profª Drª Antonella Carvalho de Oliveira

**Assistentes Editoriais**

Natalia Oliveira

Bruno Oliveira

Flávia Roberta Barão

**Bibliotecário**

Maurício Amormino Júnior

**Projeto Gráfico e Diagramação**

Natália Sandrini de Azevedo

Camila Alves de Cremona

Karine de Lima Wisniewski

Luiza Alves Batista

Maria Alice Pinheiro

**Imagens da Capa**

Shutterstock

**Edição de Arte**

Luiza Alves Batista

**Revisão**

Os Autores

2020 by Atena Editora

Copyright © Atena Editora

Copyright do Texto © 2020 Os autores

Copyright da Edição © 2020 Atena Editora

Direitos para esta edição cedidos à Atena Editora pelos autores.



Todo o conteúdo deste livro está licenciado sob uma Licença de Atribuição *Creative Commons*. Atribuição 4.0 Internacional (CC BY 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.

A Atena Editora não se responsabiliza por eventuais mudanças ocorridas nos endereços convencionais ou eletrônicos citados nesta obra.

Todos os manuscritos foram previamente submetidos à avaliação cega pelos pares, membros do Conselho Editorial desta Editora, tendo sido aprovados para a publicação.

**Conselho Editorial**

**Ciências Humanas e Sociais Aplicadas**

Prof. Dr. Álvaro Augusto de Borba Barreto – Universidade Federal de Pelotas

Prof. Dr. Alexandre Jose Schumacher – Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso

Prof. Dr. Américo Junior Nunes da Silva – Universidade do Estado da Bahia

Prof. Dr. Antonio Carlos Frasson – Universidade Tecnológica Federal do Paraná

Prof. Dr. Antonio Gasparetto Júnior – Instituto Federal do Sudeste de Minas Gerais

Prof. Dr. Antonio Isidro-Filho – Universidade de Brasília

Prof. Dr. Carlos Antonio de Souza Moraes – Universidade Federal Fluminense

Profª Drª Cristina Gaio – Universidade de Lisboa

Prof. Dr. Deyvison de Lima Oliveira – Universidade Federal de Rondônia  
Prof. Dr. Edvaldo Antunes de Farias – Universidade Estácio de Sá  
Prof. Dr. Elson Ferreira Costa – Universidade do Estado do Pará  
Prof. Dr. Eloi Martins Senhora – Universidade Federal de Roraima  
Prof. Dr. Gustavo Henrique Cepolini Ferreira – Universidade Estadual de Montes Claros  
Profª Drª Ivone Goulart Lopes – Istituto Internazionele delle Figlie de Maria Ausiliatrice  
Prof. Dr. Jadson Correia de Oliveira – Universidade Católica do Salvador  
Prof. Dr. Julio Candido de Meirelles Junior – Universidade Federal Fluminense  
Profª Drª Lina Maria Gonçalves – Universidade Federal do Tocantins  
Prof. Dr. Luis Ricardo Fernandes da Costa – Universidade Estadual de Montes Claros  
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
Prof. Dr. Marcelo Pereira da Silva – Pontifícia Universidade Católica de Campinas  
Profª Drª Maria Luzia da Silva Santana – Universidade Federal de Mato Grosso do Sul  
Profª Drª Paola Andressa Scortegagna – Universidade Estadual de Ponta Grossa  
Profª Drª Rita de Cássia da Silva Oliveira – Universidade Estadual de Ponta Grossa  
Prof. Dr. Rui Maia Diamantino – Universidade Salvador  
Prof. Dr. Urandi João Rodrigues Junior – Universidade Federal do Oeste do Pará  
Profª Drª Vanessa Bordin Viera – Universidade Federal de Campina Grande  
Prof. Dr. William Cleber Domingues Silva – Universidade Federal Rural do Rio de Janeiro  
Prof. Dr. Willian Douglas Guilherme – Universidade Federal do Tocantins

#### **Ciências Agrárias e Multidisciplinar**

Prof. Dr. Alexandre Igor Azevedo Pereira – Instituto Federal Goiano  
Profª Drª Carla Cristina Bauermann Brasil – Universidade Federal de Santa Maria  
Prof. Dr. Antonio Pasqualetto – Pontifícia Universidade Católica de Goiás  
Prof. Dr. Cleberton Correia Santos – Universidade Federal da Grande Dourados  
Profª Drª Daiane Garabeli Trojan – Universidade Norte do Paraná  
Profª Drª Diocléa Almeida Seabra Silva – Universidade Federal Rural da Amazônia  
Prof. Dr. Écio Souza Diniz – Universidade Federal de Viçosa  
Prof. Dr. Fábio Steiner – Universidade Estadual de Mato Grosso do Sul  
Prof. Dr. Fágner Cavalcante Patrocínio dos Santos – Universidade Federal do Ceará  
Profª Drª Girlene Santos de Souza – Universidade Federal do Recôncavo da Bahia  
Prof. Dr. Jael Soares Batista – Universidade Federal Rural do Semi-Árido  
Prof. Dr. Júlio César Ribeiro – Universidade Federal Rural do Rio de Janeiro  
Profª Drª Lina Raquel Santos Araújo – Universidade Estadual do Ceará  
Prof. Dr. Pedro Manuel Villa – Universidade Federal de Viçosa  
Profª Drª Raissa Rachel Salustriano da Silva Matos – Universidade Federal do Maranhão  
Prof. Dr. Ronilson Freitas de Souza – Universidade do Estado do Pará  
Profª Drª Talita de Santos Matos – Universidade Federal Rural do Rio de Janeiro  
Prof. Dr. Tiago da Silva Teófilo – Universidade Federal Rural do Semi-Árido  
Prof. Dr. Valdemar Antonio Paffaro Junior – Universidade Federal de Alfenas

#### **Ciências Biológicas e da Saúde**

Prof. Dr. André Ribeiro da Silva – Universidade de Brasília  
Profª Drª Anelise Levay Murari – Universidade Federal de Pelotas  
Prof. Dr. Benedito Rodrigues da Silva Neto – Universidade Federal de Goiás  
Prof. Dr. Douglas Siqueira de Almeida Chaves – Universidade Federal Rural do Rio de Janeiro

Prof. Dr. Edson da Silva – Universidade Federal dos Vales do Jequitinhonha e Mucuri  
Prof<sup>a</sup> Dr<sup>a</sup> Eleuza Rodrigues Machado – Faculdade Anhanguera de Brasília  
Prof<sup>a</sup> Dr<sup>a</sup> Elane Schwinden Prudêncio – Universidade Federal de Santa Catarina  
Prof<sup>a</sup> Dr<sup>a</sup> Eysler Gonçalves Maia Brasil – Universidade da Integração Internacional da Lusofonia Afro-Brasileira  
Prof. Dr. Ferlando Lima Santos – Universidade Federal do Recôncavo da Bahia  
Prof<sup>a</sup> Dr<sup>a</sup> Gabriela Vieira do Amaral – Universidade de Vassouras  
Prof. Dr. Gianfábio Pimentel Franco – Universidade Federal de Santa Maria  
Prof. Dr. Helio Franklin Rodrigues de Almeida – Universidade Federal de Rondônia  
Prof<sup>a</sup> Dr<sup>a</sup> Iara Lúcia Tescarollo – Universidade São Francisco  
Prof. Dr. Igor Luiz Vieira de Lima Santos – Universidade Federal de Campina Grande  
Prof. Dr. Jesus Rodrigues Lemos – Universidade Federal do Piauí  
Prof. Dr. Jônatas de França Barros – Universidade Federal do Rio Grande do Norte  
Prof. Dr. José Max Barbosa de Oliveira Junior – Universidade Federal do Oeste do Pará  
Prof. Dr. Luís Paulo Souza e Souza – Universidade Federal do Amazonas  
Prof<sup>a</sup> Dr<sup>a</sup> Magnólia de Araújo Campos – Universidade Federal de Campina Grande  
Prof. Dr. Marcus Fernando da Silva Praxedes – Universidade Federal do Recôncavo da Bahia  
Prof<sup>a</sup> Dr<sup>a</sup> Mylena Andréa Oliveira Torres – Universidade Ceuma  
Prof<sup>a</sup> Dr<sup>a</sup> Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
Prof. Dr. Paulo Inada – Universidade Estadual de Maringá  
Prof<sup>a</sup> Dr<sup>a</sup> Regiane Luz Carvalho – Centro Universitário das Faculdades Associadas de Ensino  
Prof<sup>a</sup> Dr<sup>a</sup> Renata Mendes de Freitas – Universidade Federal de Juiz de Fora  
Prof<sup>a</sup> Dr<sup>a</sup> Vanessa Lima Gonçalves – Universidade Estadual de Ponta Grossa  
Prof<sup>a</sup> Dr<sup>a</sup> Vanessa Bordin Viera – Universidade Federal de Campina Grande

#### **Ciências Exatas e da Terra e Engenharias**

Prof. Dr. Adélio Alcino Sampaio Castro Machado – Universidade do Porto  
Prof. Dr. Alexandre Leite dos Santos Silva – Universidade Federal do Piauí  
Prof. Dr. Carlos Eduardo Sanches de Andrade – Universidade Federal de Goiás  
Prof<sup>a</sup> Dr<sup>a</sup> Carmen Lúcia Voigt – Universidade Norte do Paraná  
Prof. Dr. Douglas Gonçalves da Silva – Universidade Estadual do Sudoeste da Bahia  
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná  
Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará  
Prof<sup>a</sup> Dra. Jéssica Verger Nardeli – Universidade Estadual Paulista Júlio de Mesquita Filho  
Prof. Dr. Juliano Carlo Rufino de Freitas – Universidade Federal de Campina Grande  
Prof<sup>a</sup> Dr<sup>a</sup> 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<sup>a</sup> Dr<sup>a</sup> Neiva Maria de Almeida – Universidade Federal da Paraíba  
Prof<sup>a</sup> Dr<sup>a</sup> Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista

#### **Linguística, Letras e Artes**

Prof<sup>a</sup> Dr<sup>a</sup> Adriana Demite Stephani – Universidade Federal do Tocantins  
Prof<sup>a</sup> Dr<sup>a</sup> Angeli Rose do Nascimento – Universidade Federal do Estado do Rio de Janeiro  
Prof<sup>a</sup> Dr<sup>a</sup> Carolina Fernandes da Silva Mandaji – Universidade Tecnológica Federal do Paraná  
Prof<sup>a</sup> Dr<sup>a</sup> Denise Rocha – Universidade Federal do Ceará

Prof. Dr. Fabiano Tadeu Grazioli – Universidade Regional Integrada do Alto Uruguai e das Missões  
Prof. Dr. Gilmei Fleck – Universidade Estadual do Oeste do Paraná  
Profª Drª Keyla Christina Almeida Portela – Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso  
Profª Drª Miranilde Oliveira Neves – Instituto de Educação, Ciência e Tecnologia do Pará  
Profª Drª Sandra Regina Gardacho Pietrobon – Universidade Estadual do Centro-Oeste  
Profª Drª Sheila Marta Carregosa Rocha – Universidade do Estado da Bahia

#### **Conselho Técnico Científico**

Prof. Me. Abrãao Carvalho Nogueira – Universidade Federal do Espírito Santo  
Prof. Me. Adalberto Zorzo – Centro Estadual de Educação Tecnológica Paula Souza  
Prof. Me. Adalto Moreira Braz – Universidade Federal de Goiás  
Prof. Dr. Adaylson Wagner Sousa de Vasconcelos – Ordem dos Advogados do Brasil/Seccional Paraíba  
Prof. Dr. Adilson Tadeu Basquerote Silva – Universidade para o Desenvolvimento do Alto Vale do Itajaí  
Prof. Me. Alexsandro Teixeira Ribeiro – Centro Universitário Internacional  
Prof. Me. André Flávio Gonçalves Silva – Universidade Federal do Maranhão  
Profª Ma. Anne Karynne da Silva Barbosa – Universidade Federal do Maranhão  
Profª Drª Andreza Lopes – Instituto de Pesquisa e Desenvolvimento Acadêmico  
Profª Drª Andrezza Miguel da Silva – Faculdade da Amazônia  
Prof. Dr. Antonio Hot Pereira de Faria – Polícia Militar de Minas Gerais  
Prof. Me. Armando Dias Duarte – Universidade Federal de Pernambuco  
Profª Ma. Bianca Camargo Martins – UniCesumar  
Profª Ma. Carolina Shimomura Nanya – Universidade Federal de São Carlos  
Prof. Me. Carlos Antônio dos Santos – Universidade Federal Rural do Rio de Janeiro  
Prof. Ma. Cláudia de Araújo Marques – Faculdade de Música do Espírito Santo  
Profª Drª Cláudia Taís Siqueira Cagliari – Centro Universitário Dinâmica das Cataratas  
Prof. Me. Daniel da Silva Miranda – Universidade Federal do Pará  
Profª Ma. Daniela da Silva Rodrigues – Universidade de Brasília  
Profª Ma. Daniela Remião de Macedo – Universidade de Lisboa  
Profª Ma. Dayane de Melo Barros – Universidade Federal de Pernambuco  
Prof. Me. Douglas Santos Mezacas – Universidade Estadual de Goiás  
Prof. Me. Edevaldo de Castro Monteiro – Embrapa Agrobiologia  
Prof. Me. Eduardo Gomes de Oliveira – Faculdades Unificadas Doctum de Cataguases  
Prof. Me. Eduardo Henrique Ferreira – Faculdade Pitágoras de Londrina  
Prof. Dr. Edwaldo Costa – Marinha do Brasil  
Prof. Me. Eliel Constantino da Silva – Universidade Estadual Paulista Júlio de Mesquita  
Prof. Me. Ernane Rosa Martins – Instituto Federal de Educação, Ciência e Tecnologia de Goiás  
Prof. Me. Eivaldo de Sousa Costa Junior – Prefeitura Municipal de São João do Piauí  
Profª Ma. Fabiana Coelho Couto Rocha Corrêa – Centro Universitário Estácio Juiz de Fora  
Prof. Dr. Fabiano Lemos Pereira – Prefeitura Municipal de Macaé  
Prof. Me. Felipe da Costa Negrão – Universidade Federal do Amazonas  
Profª Drª Germana Ponce de Leon Ramírez – Centro Universitário Adventista de São Paulo  
Prof. Me. Gevair Campos – Instituto Mineiro de Agropecuária  
Prof. Dr. Guilherme Renato Gomes – Universidade Norte do Paraná  
Prof. Me. Gustavo Krahl – Universidade do Oeste de Santa Catarina  
Prof. Me. Helton Rangel Coutinho Junior – Tribunal de Justiça do Estado do Rio de Janeiro  
Profª Ma. Isabelle Cerqueira Sousa – Universidade de Fortaleza

Profª Ma. Jaqueline Oliveira Rezende – Universidade Federal de Uberlândia  
Prof. Me. Javier Antonio Albornoz – University of Miami and Miami Dade College  
Prof. Me. Jhonatan da Silva Lima – Universidade Federal do Pará  
Prof. Dr. José Carlos da Silva Mendes – Instituto de Psicologia Cognitiva, Desenvolvimento Humano e Social  
Prof. Me. Jose Elyton Batista dos Santos – Universidade Federal de Sergipe  
Prof. Me. José Luiz Leonardo de Araujo Pimenta – Instituto Nacional de Investigación Agropecuaria Uruguay  
Prof. Me. José Messias Ribeiro Júnior – Instituto Federal de Educação Tecnológica de Pernambuco  
Profª Drª Juliana Santana de Curcio – Universidade Federal de Goiás  
Profª Ma. Juliana Thaisa Rodrigues Pacheco – Universidade Estadual de Ponta Grossa  
Profª Drª Kamilly Souza do Vale – Núcleo de Pesquisas Fenomenológicas/UFGA  
Prof. Dr. Kárpio Márcio de Siqueira – Universidade do Estado da Bahia  
Profª Drª Karina de Araújo Dias – Prefeitura Municipal de Florianópolis  
Prof. Dr. Lázaro Castro Silva Nascimento – Laboratório de Fenomenologia & Subjetividade/UFPR  
Prof. Me. Leonardo Tullio – Universidade Estadual de Ponta Grossa  
Profª Ma. Lilian Coelho de Freitas – Instituto Federal do Pará  
Profª Ma. Liliani Aparecida Sereno Fontes de Medeiros – Consórcio CEDERJ  
Profª Drª Lívia do Carmo Silva – Universidade Federal de Goiás  
Prof. Me. Lucio Marques Vieira Souza – Secretaria de Estado da Educação, do Esporte e da Cultura de Sergipe  
Prof. Me. Luis Henrique Almeida Castro – Universidade Federal da Grande Dourados  
Prof. Dr. Luan Vinicius Bernardelli – Universidade Estadual do Paraná  
Prof. Dr. Michel da Costa – Universidade Metropolitana de Santos  
Prof. Dr. Marcelo Máximo Purificação – Fundação Integrada Municipal de Ensino Superior  
Prof. Me. Marcos Aurelio Alves e Silva – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo  
Profª Ma. Maria Elanny Damasceno Silva – Universidade Federal do Ceará  
Profª Ma. Marileila Marques Toledo – Universidade Federal dos Vales do Jequitinhonha e Mucuri  
Prof. Me. Ricardo Sérgio da Silva – Universidade Federal de Pernambuco  
Prof. Me. Rafael Henrique Silva – Hospital Universitário da Universidade Federal da Grande Dourados  
Profª Ma. Renata Luciane Polsaque Young Blood – UniSecal  
Prof. Me. Sebastião André Barbosa Junior – Universidade Federal Rural de Pernambuco  
Profª Ma. Silene Ribeiro Miranda Barbosa – Consultoria Brasileira de Ensino, Pesquisa e Extensão  
Profª Ma. Solange Aparecida de Souza Monteiro – Instituto Federal de São Paulo  
Prof. Me. Tallys Newton Fernandes de Matos – Faculdade Regional Jaguaribana  
Profª Ma. Thatianny Jasmine Castro Martins de Carvalho – Universidade Federal do Piauí  
Prof. Me. Tiago Silvio Dedoné – Colégio ECEL Positivo  
Prof. Dr. Welleson Feitosa Gazel – Universidade Paulista

# Sistemas de informação e aplicações computacionais

**Editora Chefe:** Profª Drª Antonella Carvalho de Oliveira  
**Bibliotecário:** Maurício Amormino Júnior  
**Diagramação:** Karine de Lima Wisniewski  
**Edição de Arte:** Luiza Alves Batista  
**Revisão:** Os Autores  
**Organizador:** Ernane Rosa Martins

## Dados Internacionais de Catalogação na Publicação (CIP) (eDOC BRASIL, Belo Horizonte/MG)

S622 Sistemas de informação e aplicações computacionais [recurso eletrônico] / Organizador Ernane Rosa Martins. – Ponta Grossa, PR: Atena, 2020.

Formato: PDF

Requisitos de sistema: Adobe Acrobat Reader

Modo de acesso: World Wide Web

Inclui bibliografia

ISBN 978-65-5706-317-0

DOI 10.22533/at.ed.170201808

1. Computação – Pesquisa – Brasil. I. Martins, Ernane Rosa.  
CDD 004

**Elaborado por Maurício Amormino Júnior – CRB6/2422**

**Atena Editora**

Ponta Grossa – Paraná – Brasil

Telefone: +55 (42) 3323-5493

[www.atenaeditora.com.br](http://www.atenaeditora.com.br)

[contato@atenaeditora.com.br](mailto:contato@atenaeditora.com.br)

## APRESENTAÇÃO

O termo Sistemas de Informação (SI), é utilizado para descrever sistemas que sejam automatizados. Este campo de estudo se preocupa com questões, tais como: o desenvolvimento, uso e implicações das tecnologias de informação e comunicação nas organizações. Os dados são os fatos de forma bruta das organizações, antes de terem sido organizados e arranjados de forma que as pessoas os entendam e possam usá-los. As informações, por sua vez, são os dados de forma significativa e útil para as pessoas.

Dentro deste contexto, esta obra aborda diversos assuntos relevantes para profissionais e estudantes das mais diversas áreas, tais como: um sistema para automatizar o processo de seleção de alunos, a investigação da visão computacional para classificar automaticamente a modalidade de uma imagem médica, o projeto extensionista “Clube de programação e robótica”, as estratégias do framework MeteorJS para a sincronização de dados entre os clientes e os servidores, a proposta de um modelo de predição capaz de identificar perfis de condução de motoristas utilizando aprendizado de máquina, a avaliação das estratégias, arquiteturas e metodologia aplicadas na Integração de aplicativos nos processos de gestão e organização da informação, o desenvolvimento de um jogo educativo, para auxiliar o processo de ensino-aprendizagem na área de testes de software, um ensaio que apresenta um método baseado nos RF-CC-17, para elaborar um Mapeamento de Conformidade e Mobilização (MCM), a análise das estratégias do modelo pedagógico ML-SAI, o qual foi desenvolvido para orientar atividades de m-learning, fundamentado na Teoria da Sala de Aula Invertida (SAI), uma proposta de um método para o projeto, a fabricação e o teste de um veículo aéreo não tripulado de baixo custo, o uso de dois modelos neurais trabalhando em conjunto a fim de efetuar a tarefa de detecção de pedestres, rastreamento e contagem por meio de imagens digitais, um estudo sobre a segurança em redes sociais, um sistema de elicitação de requisitos orientado pela modelagem de processo de negócio, um Sistema de Informação Ambiental, desenvolvido para armazenar e permitir a consulta de dados históricos ambientais, o uso de técnicas para segurança em aplicações web, uma metodologia que possa aumentar a confiança dos dados na entrada e saída do dinheiro público com uma rede blockchain, a construção de um simulador do reator nuclear de pesquisa TRIGA IPR-R1.

Sendo assim, os trabalhos que compõe esta obra permitem aos seus leitores, analisar e discutir os diversos assuntos interessantes abordados. Por fim, desejamos a cada autor, nossos mais sinceros agradecimentos por suas contribuições, e aos leitores, desejamos uma excelente leitura com excelentes e novas reflexões.

Ernane Rosa Martins

## SUMÁRIO

<b>CAPÍTULO 1</b> .....	<b>1</b>
AUTOMAÇÃO DE PROCESSOS DA SECRETARIA GERAL DE UNIVERSIDADES VISANDO A SUSTENTABILIDADE	
Beatriz da Mota Bonanno Daniela Vieira Cunha Fabio Kawaoka Takase	
<b>DOI 10.22533/at.ed.1702018081</b>	
<b>CAPÍTULO 2</b> .....	<b>15</b>
CLASSIFICAÇÃO DE IMAGENS MÉDICAS EM MODALIDADES USANDO VISÃO COMPUTACIONAL	
Sara Conceição de Sousa Araújo Silva Glauco Vitor Pedrosa	
<b>DOI 10.22533/at.ed.1702018082</b>	
<b>CAPÍTULO 3</b> .....	<b>26</b>
CLUBE DE PROGRAMAÇÃO E ROBÓTICA: EXPERIMENTOS EDUCACIONAIS NO ENSINO FUNDAMENTAL NO INTERIOR DA AMAZÔNIA	
Ruan Carlos Tavares Reis Andrew Pedreiro Amorim Angel Pena Galvão Andrik Guimarães Ferreira Juarez Benedito da Silva Clayton André Maia dos Santos	
<b>DOI 10.22533/at.ed.1702018083</b>	
<b>CAPÍTULO 4</b> .....	<b>36</b>
ESTRATÉGIAS PARA SINCRONIZAÇÃO E PROTEÇÃO DE DADOS EM APLICAÇÕES WEB REAL-TIME UTILIZANDO METEORJS	
Renan Gomes Barreto Lucas Oliveira Costa Aversari	
<b>DOI 10.22533/at.ed.1702018084</b>	
<b>CAPÍTULO 5</b> .....	<b>48</b>
IDENTIFICAÇÃO AUTOMÁTICA DE PERFIS DE MOTORISTAS USANDO APRENDIZADO DE MÁQUINA	
Ricardo Roberto Carlos da Silva Júnior Hilário Tomaz Alves de Oliveira	
<b>DOI 10.22533/at.ed.1702018085</b>	
<b>CAPÍTULO 6</b> .....	<b>60</b>
INTEGRAÇÃO DE APLICATIVOS: ESTRATÉGIA, ARQUITETURA E METODOLOGIA	
Francisco Carlos Paletta	
<b>DOI 10.22533/at.ed.1702018086</b>	
<b>CAPÍTULO 7</b> .....	<b>70</b>
ISLANDTEST: JOGO EDUCATIVO PARA APOIAR O PROCESSO ENSINO-APRENDIZAGEM DE TESTES DE SOFTWARE	
Rafael Jesus de Queiroz Fabrício de Sousa Pinto Paulo Caetano da Silva	
<b>DOI 10.22533/at.ed.1702018087</b>	

**CAPÍTULO 8 ..... 82**

MÉTODO BASEADO NOS REFERENCIAIS DE FORMAÇÃO DA SBC PARA REESTRUTURAÇÃO DE DESCRITIVOS DE DISCIPLINAS DE CIÊNCIA DA COMPUTAÇÃO EM CONFORMIDADE COM AS DCN DE 2016

Alcides Calsavara  
Ana Paula Gonçalves Serra  
Francisco de Assis Zampiroli  
Leandro Silva Galvão de Carvalho  
Miguel Jonathan  
Ronaldo Celso Messias Correia

**DOI 10.22533/at.ed.1702018088**

**CAPÍTULO 9 ..... 95**

ML-SAI: UM MODELO PEDAGÓGICO PARA ATIVIDADES DE M-LEARNING QUE INTEGRA A ABORDAGEM DA SALA DE AULA INVERTIDA

Ernane Rosa Martins  
Luís Manuel Borges Gouveia

**DOI 10.22533/at.ed.1702018089**

**CAPÍTULO 10 ..... 107**

MODELAGEM PARA ESTIMATIVA E PROJEÇÃO DE ESTOQUE DE CARBONO EM FRAGMENTOS DE FLORESTA OMBRÓFILA DENSA DE TERRAS BAIXAS POR MEIO DE ÍNDICES DE VEGETAÇÃO E LINGUAGEM R

Eric Bem dos Santos  
Hernande Pereira da Silva  
Jones Oliveira de Albuquerque

**DOI 10.22533/at.ed.17020180810**

**CAPÍTULO 11 ..... 120**

PROJETO, CONSTRUÇÃO DE UM VEÍCULO AÉREO NÃO TRIPULADO COM BASE EM CO-PROJETO DE HARDWARE E SOFTWARE

Alex Ribeiro Souza  
Mariana Cardoso  
Junio Horniche  
Patricia Boff  
João Guilherme Bonilha Viana  
Maurício Acconcia Dias

**DOI 10.22533/at.ed.17020180811**

**CAPÍTULO 12 ..... 133**

RASTREAMENTO E CONTAGEM DE PEDESTRE EM TEMPO REAL POR MEIO DE IMAGENS DIGITAIS

Alexssandro Ferreira Cordeiro  
Cristhian Urunaga Ojeda  
Pedro Luiz de Paula Filho  
Gustavo Rafael Valiati

**DOI 10.22533/at.ed.17020180812**

**CAPÍTULO 13 ..... 143**

SEGURANÇA EM REDES SOCIAIS: UMA ABORDAGEM BASEADA NA CONSCIENTIZAÇÃO DE CRIANÇAS E ADOLESCENTES EM UMA ESCOLA MUNICIPAL DA CIDADE E SANTARÉM

Clayton André Maia dos Santos  
João Vitor Mota dos Santos  
Yan Marcos Bentes dos Anjos  
Angel Pena Galvão

Irley Monteiro Araújo  
Juarez Benedito da Silva  
Aloísio Costa Barros  
Pablo Nunes de Oliveira  
Brenda da Silva Nunes

**DOI 10.22533/at.ed.17020180813**

**CAPÍTULO 14 ..... 151**

SISREMO – SISTEMA DE ELICITAÇÃO DE REQUISITOS COM BASE NA TÉCNICA REMO

Carlos Ricardo Bandeira de Souza  
Sérgio Roberto Costa Vieira

**DOI 10.22533/at.ed.17020180814**

**CAPÍTULO 15 ..... 166**

SISTEMA DE INFORMAÇÃO AMBIENTAL: VISUALIZAÇÃO DE DADOS DO ÍNDICE DE QUALIDADE DA ÁGUA APLICADO A MÚLTIPLOS PONTOS

Vania Elisabete Schneider  
Odacir Deonísio Gracioli  
Helena Graziottin Ribeiro  
Adriano Gomes da Silva  
Mayara Cechinato  
Taison Anderson Bortolin

**DOI 10.22533/at.ed.17020180815**

**CAPÍTULO 16 ..... 172**

TÉCNICAS PARA SEGURANÇA EM APLICAÇÕES WEB - BASEADO EM MESSAGE-DIGEST ALGORITHM

Daniel Rodrigues Ferraz Izario  
Yuzo Iano  
João Luiz Brancalhone Filho  
Karine Mendes Siqueira Rodrigues Ferraz Izario

**DOI 10.22533/at.ed.17020180816**

**CAPÍTULO 17 ..... 183**

UMA PROPOSTA INOVADORA UTILIZANDO BLOCKCHAIN PARA A GESTÃO FINANCEIRA EM OBRAS PÚBLICAS, TENDO COMO BASE O SISTEMA BRASILEIRO

Ricardo Silva Parente  
Ítalo Rodrigo Soares Silva  
Paulo Oliveira Siqueira Júnior  
Jorge de Almeida Brito Júnior  
Manoel Henrique Reis Nascimento  
David Barbosa de Alencar  
Jandecy Cabral Leite  
Paulo Francisco da Silva Ribeiro

**DOI 10.22533/at.ed.17020180817**

**CAPÍTULO 18 ..... 197**

UTILIZAÇÃO DO ARDUINO COMO FERRAMENTA DE DIVULGAÇÃO DA ÁREA NUCLEAR

Hudson Henrique da Silva  
Samira Santos da Silva  
Sincler Peixoto de Meireles

**DOI 10.22533/at.ed.17020180818**

**SOBRE O ORGANIZADOR..... 207**

**ÍNDICE REMISSIVO ..... 208**

## ESTRATÉGIAS PARA SINCRONIZAÇÃO E PROTEÇÃO DE DADOS EM APLICAÇÕES WEB REAL-TIME UTILIZANDO METEORJS

Data de aceite: 01/07/2020

Data de submissão: 14/05/2020

### Renan Gomes Barreto

Universidade Federal da Paraíba  
Programa de Pós-Graduação em Informática  
João Pessoa – Paraíba

ORCID: <https://orcid.org/0000-0002-0919-6957>

### Lucas Oliveira Costa Aversari

Universidade Federal de Campina Grande  
Programa de Pós-Graduação em Ciência da  
Computação

Campina Grande – Paraíba

ORCID: <https://orcid.org/0000-0003-4152-5842>

**RESUMO:** O MeteorJS é um *framework* web do tipo *full-stack* com algumas características importantes relacionadas à forma de sincronização de dados que vêm cada vez mais sendo adaptadas para uso em outros *frameworks* web. Por consequência, o estudo sobre como essas funcionalidades foram projetadas e implementadas pode ajudar na elaboração e no refinamento de outras estratégias para a sincronização de dados em aplicações web real-time. Desta maneira, o presente trabalho descreve as estratégias utilizadas pelo framework MeteorJS para a sincronização de dados entre os clientes e os servidores, apresentando os principais componentes de sua arquitetura e detalhando as

tecnologias empregadas para a sincronização de dados entre as aplicações dos lados do cliente e do servidor.

**PALAVRAS-CHAVE:** MeteorJS; Sincronização de Dados; Full-stack Framework; Banco de Dados.

### STRATEGIES FOR SYNCHRONIZATION AND DATA PROTECTION IN REAL-TIME WEB APPLICATIONS USING METEORJS

**ABSTRACT:** MeteorJS is a full-stack web framework that has some important features related to data synchronization, which is increasingly being adapted to be used by other web frameworks. Consequently, the study of how these functionalities were designed and implemented can help in the elaboration and refinement of other strategies for data synchronization in real-time web applications. Thus, the present work describes the data synchronization strategies between clients and servers used by the MeteorJS framework, presenting the main components of its architecture and detailing the technologies used for data synchronization between client and server applications.

**KEYWORDS:** MeteorJS; Data Synchronization; Full-stack Framework; Databases.

## 1 | INTRODUÇÃO

O Meteor, também conhecido como MeteorJS, é um *framework* web gratuito, livre e de código aberto. Esse *framework* foi escrito usando o Node.js, um runtime web para aplicações do lado do servidor, em que os desenvolvedores podem escrever novos módulos, estendendo funcionalidades (STRACK, 2015a).

O conceito do Meteor vai além da definição de *framework* convencional. Ele funciona como uma plataforma completa que envolve tecnologias como linguagem de programação, banco de dados e serviços web, sendo classificado como um *framework* do tipo *full-stack*. Em comparação com o AngularJS (GOOGLE, 2019), outro *framework* amplamente usado para integrar serviços e regras de negócio no lado *front-end* da aplicação, o Meteor se diferencia por conseguir lidar com a mesma lógica tanto no lado do cliente quanto no lado do servidor da aplicação.

Como herança do Node.js (OPENJS FOUNDATION, 2020), o Meteor possui ferramentas para rápida prototipagem e produção eficiente de código multiplataforma, fazendo com que as aplicações desenvolvidas com ele sejam executáveis nos mais diversos tipos de sistemas, desde Web até Android ou iOS.

A principal linguagem de programação usada para desenvolver com o Meteor é o JavaScript. Uma das grandes vantagens oferecidas pelo Meteor é a sua integração com o MongoDB, que é um banco de dados com abordagem baseada em documentos gratuitos e de código aberto. Essa estrutura do banco de dados torna desnecessária sua definição nos termos das tradicionais tabelas relacionais, bem como sua consulta por comandos SQL (STRACK, 2015a).

Essas características fazem com que o Meteor consiga integrar dados às aplicações de maneira mais simples e rápida, usando o Distributed Data Protocol (DDP), que propaga automaticamente mudanças nos dados para o lado dos clientes sem que sejam necessários stubs de sincronização projetados de forma externa (DEBERGALIS, 2012). O estudo de como essas funcionalidades foram implementadas pode ajudar na elaboração e no refinamento de outras estratégias para sincronização de dados em aplicações *web real-time*.

Sendo assim, o presente trabalho visa a descrever as estratégias de sincronização de dados entre as aplicações nos lados do cliente e do servidor utilizadas pelo *framework* MeteorJS. Desta forma, foram definidos como objetivos específicos, em sequência:

- Apresentar a arquitetura de um aplicativo Meteor e seus principais componentes;
- Descrever as estratégias utilizadas pelo *framework* para transferência de dados entre os diferentes componentes do sistema.

Nas próximas seções serão apresentadas a fundamentação teórica, a metodologia e, por fim, as considerações finais desta pesquisa.

## 2 | FUNDAMENTAÇÃO TEÓRICA

### 2.1 Aplicações web baseadas em JavaScript

O JavaScript é uma linguagem de programação interpretada. Foi desenvolvida para que scripts fossem executados no lado do cliente e interagissem com o usuário sem que fossem manipulados pelo servidor. É orientada a objetos, baseada em protótipos, e de tipagem dinâmica e fraca. Possui recursos avançados como o suporte à programação funcional, recursos de fechamento e funções de alta ordem, que normalmente não estão disponíveis em linguagens como Java e C++ (TILKOV; VINOSKI, 2010).

A linguagem JavaScript foi desenvolvida por Brendan Eich na Netscape. Inicialmente, a linguagem chamava-se Mocha e posteriormente teve seu nome mudado para LiveScript, que, eventualmente, tornou-se a linguagem de programação JavaScript. Apesar de ter sido lançada com o nome de LiveScript numa versão beta do navegador Netscape, teve seu nome definitivamente alterado num anúncio conjunto com a Sun Microsystems para JavaScript (FLANAGAN; MATILAINEN, 2007). O fato de o nome ter sido mudado para JavaScript coincidiu com o período em que o Netscape começou a aceitar scripts Java (*applets*), fazendo com que o nome escolhido gerasse certa confusão, pois as pessoas pensavam que a linguagem tinha sido baseada em Java. Alguns dizem que tal escolha foi por motivos de marketing, para se aproveitar da linguagem recém-lançada Java.

JavaScript é considerada a linguagem mais popular para programação web. O potencial da linguagem demorou a ser descoberto, já que sua sintaxe era voltada para o público leigo. Isto superado, ocorreu a proliferação de *frameworks* e bibliotecas, práticas de programação melhoradas e aumento no uso do JavaScript fora do ambiente de navegadores, bem como o uso de plataformas de JavaScript *server-side*. No ano de 2009, a linguagem foi finalmente padronizada por meio da biblioteca CommonJS (FLANAGAN; MATILAINEN, 2007).

### 2.2 AngularJS

O AngularJS é um dos frameworks JavaScript mais utilizados atualmente. Ele é mantido pelo Google e faz parte de uma iniciativa open-source para desenvolver aplicações *single-page*. O objetivo do Angular é facilitar o acesso a aplicativos por meio do navegador web, além de facilitar os testes, já que foi desenvolvido em cima do padrão *model-view-viewmodel* (MVVM) (GOOGLE, 2019).

O padrão MVVM facilita a separação do desenvolvimento da interface gráfica do desenvolvimento da lógica de controle que está por trás dela (Figura 1). O *view model* atua como um conversor de valores, ou seja, ele é responsável por converter os dados das classes de modelagem de forma que eles fiquem facilmente apresentáveis pelas classes do *front-end* (*view*). Ou seja, além de lidar com os dados, o *view model* também controla sua exibição e coordena o *front-end* (TIMMS, 2014).

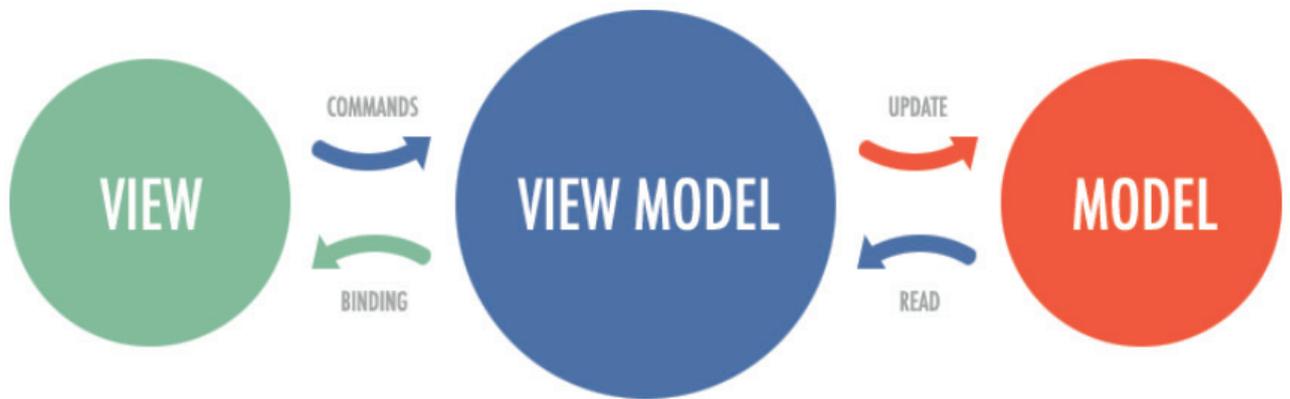


Figura 1: Arquitetura do padrão MVVM. Fonte: Adaptado de Bastos (2017).

O AngularJS foi desenvolvido com base no fato de que a programação declarativa é melhor que a programação imperativa para se construir interfaces com o usuário e conectar módulos de software. Enquanto isso, a programação imperativa se mantém como via de regra para a escrita de modelos de negócio (*back-end*). Baseado nesse princípio, o Angular se adapta e estende o HTML de forma que conteúdos dinâmicos sejam melhor gerenciados, com uma ligação direta e bidimensional entre *model* e *view*, sincronizando automaticamente mudanças provocadas por uma das camadas na outra, por meio de uma abstração do DOM (*data-object-model*), ou seja, da camada de dados (GOOGLE, 2019). Tal ligação bidimensional é sua principal característica e reduz drasticamente a quantidade de código necessária para exibir dados processados pelo servidor no lado do cliente. *Templates* em HTML são renderizados em tempo real sobre dados contidos num escopo delimitado pela tag “*\$scope*”, que detecta mudanças no modelo dos dados e atualiza o HTML do *view* por meio de um script predefinido.

A abstração dos dados da lógica do aplicativo melhora os testes do código. Além disso, os testes se tornam tão importantes quanto o desenvolvimento, sendo facilitados pelo baixo acoplamento entre o lado do cliente e o lado do servidor, permitindo que o desenvolvimento cresça paralelamente de ambos os lados e que o código seja reutilizado (FLANAGAN; MATILAINEN, 2007).

## 2.3 MeteorJS

O Meteor é uma plataforma JavaScript *full-stack*, com tecnologias padrão para cada uma das camadas da pilha, começando com o banco de dados e terminando na interface do usuário e renderização. O usuário pode optar por trocar as tecnologias padrão por outras à sua escolha – por exemplo, pode mudar o *front-end* padrão de Blaze para React ou Angular (COLEMAN; GREIF, 2015).

A história do Meteor é relativamente pequena, tendo surgido em dezembro de 2011 inicialmente com o nome de Project Skybreak, desenvolvido pela *startup* Meteor Development. Seu nome só se tornou Meteor em 2012, por razões de simplicidade, e, desde então, ele vem se popularizando e ganhando cada vez mais espaço entre os desenvolvedores web (COLEMAN; GREIF, 2015).

Em outubro de 2014, a Meteor Development adquiriu a *startup* FanthomDB com o objetivo de expandir o suporte ao banco de dados do Meteor, o que a princípio não funcionou como o esperado, mas abriu novos horizontes para que a parceria entre a MongoDB Inc. e a Meteor fosse fechada mais tarde no mesmo ano. Desde então, o Meteor assumiu uma arquitetura similar à utilizada atualmente e vem sendo constantemente atualizado, tendo passado por mais de 10 versões estáveis (COLEMAN; GREIF, 2015).

## 3 | ESTRATÉGIAS PARA SINCRONIZAÇÃO E PROTEÇÃO DE DADOS EM APLICAÇÕES WEB REAL-TIME UTILIZANDO METEORJS

O MeteorJS é um framework web do tipo *full-stack*. Algumas de suas características mais importantes vêm sendo cada vez mais adaptadas para outros frameworks.

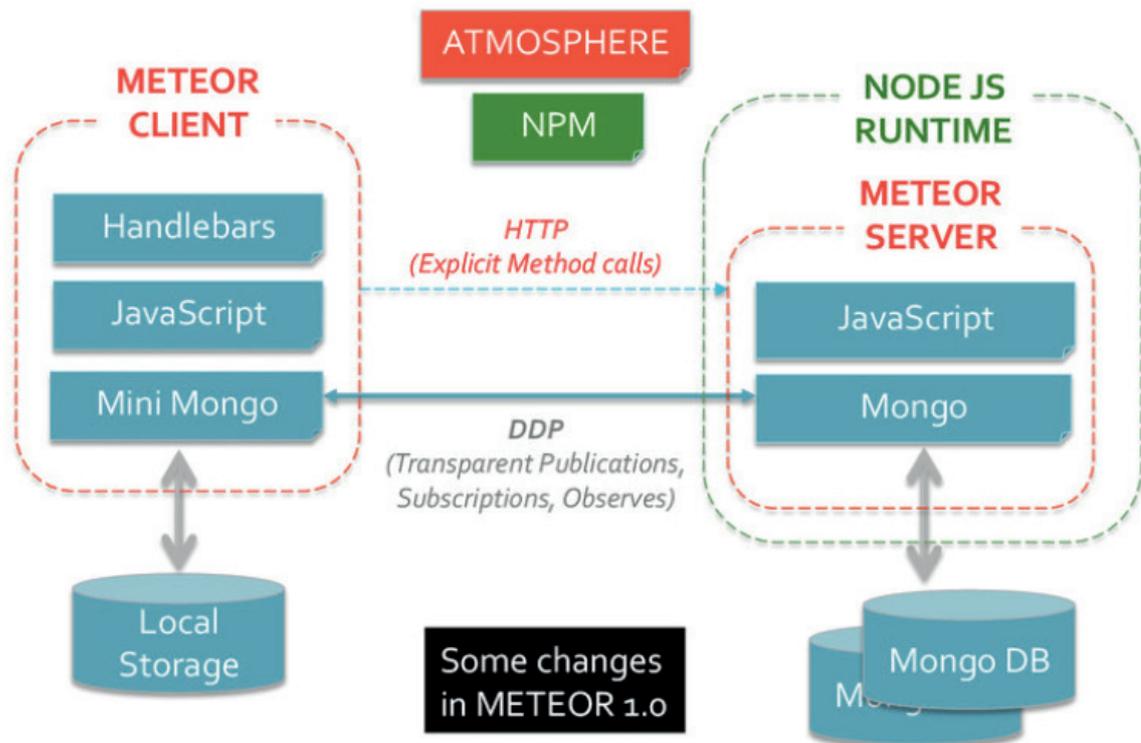


Figura 2: A figura mostra os principais componentes do Meteor nos lados cliente e servidor. Fonte: MongoDB Inc., 2015.

A Figura 2 mostra os principais componentes da arquitetura do Meteor e como as partes se comunicam. Do lado esquerdo é apresentado o aplicativo JavaScript executado no navegador do usuário. No lado direito da imagem, podemos ver a implementação do servidor da aplicação. Desta forma, nota-se que os componentes estão bem acoplados e ligados por subcomponentes como o MiniMongo e tecnologias de transferência de dados como o DDP e LiveQuery.

Durante o desenvolvimento do framework Meteor, conforme apresentado por Strack (2015a), foram seguidos alguns princípios básicos:

- **Dados sempre na rede:** O Meteor não envia qualquer dado pronto em HTML pela rede, em vez disso ele trabalha como um serviço baseado em restful, enviando os dados para o cliente e deixando com que este os renderize.
- **Uma única linguagem:** O Meteor possibilita que o usuário escreva, de maneira única em ambos os lados (cliente e servidor) da sua aplicação, na linguagem JavaScript. No lado do cliente, isso já seria um padrão, mas no servidor é uma novidade, graças ao Node.js.
- **Banco de dados onipotente:** O usuário pode acessar o banco de dados tanto no lado do servidor quanto no lado do cliente, evitando o clássico transtorno de se conectar ao banco por meio de requisições HTTP desnecessárias.
- **Compensação de latência:** O Meteor costuma pré-carregar alguns dados e modelos no lado do cliente, fazendo com que a latência seja compensada e que chamadas e consultas ao banco pareçam instantâneas ao cliente. É uma forma de cachê inteligente.

- **Atualização automática de camadas:** Como o Meteor é voltado para aplicações em tempo real, todas as camadas, desde a gráfica até o banco de dados, são atualizadas automaticamente quando necessário.
- **Simplicidade e produtividade:** O Meteor possui uma API muito limpa, com métodos de chamada simples e usando vários padrões de projeto. Conseqüentemente, a produtividade sofre um aumento intenso. Para conectar novos serviços e APIs, basta utilizar o protocolo DDP, fazendo com que sempre que uma mudança aconteça no servidor, as aplicações do lado cliente sejam notificadas e atualizadas.

### 3.1 Principais componentes da arquitetura

Uma aplicação Meteor segue uma arquitetura cliente-servidor típica, porém, a integração e a transferência de dados entre as diferentes camadas do sistema são feitas de forma transparente ao desenvolvedor, permitindo a implementação facilitada de técnicas de redução de latência, segurança e cachê.

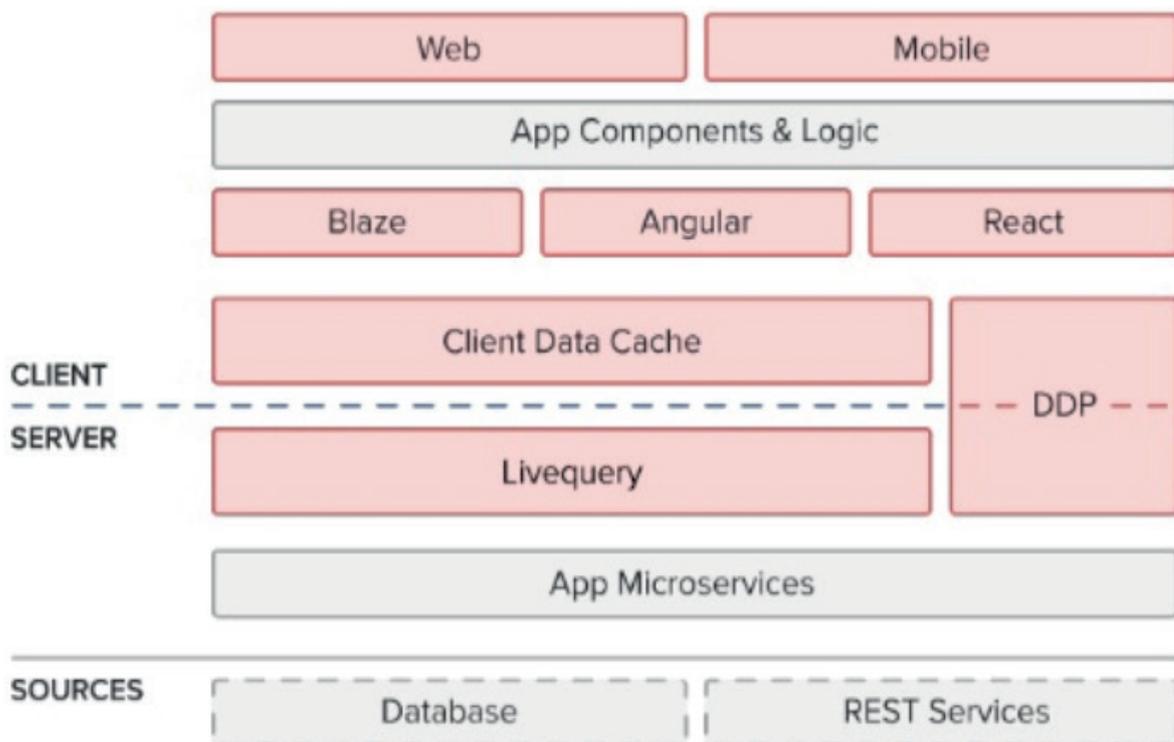


Figura 3: Principais componentes da arquitetura do Meteor. Fonte: Designveloper, 2016.

A Figura 3 mostra as diferentes camadas de uma aplicação Meteor e seus subcomponentes. Cada subcomponente tem um papel importante na transmissão, armazenamento e segurança dos dados e, de maneira geral, podem ser descritos como:

- **Web/Mobile:** Ao criar uma aplicação por meio do Meteor, o desenvolvedor poderá usar o mesmo código para gerar versões para navegadores e dispositivos móveis como Android e iOS.

- **Lógica da aplicação:** Aqui ficam os componentes e a lógica da aplicação visíveis do lado do cliente.
- **Caché de dados no lado do cliente:** O Meteor possui uma versão simplificada do banco de dados MongoDB que é chamada de MiniMongo. Esse componente é responsável por armazenar um subconjunto dos documentos da coleção principal presente no lado do servidor.
- **UI Framework:** Para prover a reatividade, o Meteor possui nativamente o *framework* Blaze. Caso necessário, o desenvolvedor poderá escolher utilizar outros *frameworks* similares como o AngularJS e o React. Todos são suportados pelo Meteor, e eles também se beneficiam do acesso facilitado aos dados que o Meteor provê.
- **Distributed Data Protocol (DDP):** Principal componente da arquitetura do Meteor. É o protocolo que o Meteor usa para a comunicação entre o cliente e o servidor. É um protocolo minimalista que tem dois objetivos: gerenciar as chamadas de procedimento remoto e atualizar os dados de forma bidirecional.
- **Live query:** O suporte a *live queries* também é um aspecto importante no Meteor. Normalmente, uma *live query* é criada quando se retorna um cursor para uma publicação. Isso permite que o desenvolvedor acesse os dados no lado do cliente como se estivesse acessando os dados no lado do servidor.
- **App microservices:** Embora seja opcional, a utilização de uma arquitetura baseada em microservices é recomendável. Usando essa estrutura, o aplicativo é separado em serviços independentes, o que torna a aplicação escalável e mais fácil de ser mantida.
- **Sources (database/rest services):** Os dados podem ser lidos de várias fontes distintas. Normalmente, as coleções do MongoDB são suficientes para desenvolver a aplicação, porém, em alguns casos, é necessária a utilização de fontes de dados externas como APIs.

### 3.2 Persistência de dados no Meteor

Uma das principais características do Meteor é a sincronização automática de informações entre o cliente e o servidor. Esse diferencial permite ao desenvolvedor ter um código mais limpo e organizado, diminuindo os custos com a manutenção do aplicativo (DESIGNVELOPER, 2016).

Por ser um *full-stack framework* desenvolvido em JavaScript, o Meteor permite o acesso ao banco diretamente do *front-end* de forma transparente. Outra característica relevante é que o Meteor possui mecanismos para fazer caché de dados e para bloquear seu acesso do lado do cliente. Isso possibilita uma interface gráfica mais fluida, transmitindo ao usuário a impressão de que os dados estão armazenados localmente.

Coleções do banco de dados podem ser acessadas diretamente como uma variável local e esse acesso simplificado aos dados permite que o desenvolvedor foque seus esforços em partes mais importantes do código, como a lógica da aplicação (STUBAILO, 2015).

A maioria dos desenvolvedores web já está familiarizada com JavaScript e, assim, não será necessário aprender nenhuma linguagem nova. Além disso, os métodos são compartilhados tanto no lado do cliente quanto no lado do servidor.

Outro ponto interessante é que a abstração do acesso aos dados no Meteor reduz consideravelmente muitos possíveis aborrecimentos e armadilhas que são comuns no desenvolvimento web.

A peça-chave para que a sincronização dos dados aconteça chama-se coleção, que será melhor detalhada na próxima subseção.

### 3.2.1 Coleções

Uma coleção é uma estrutura de dados especial que é responsável por sincronizar, em tempo real, as informações do navegador com o banco de dados MongoDB. Caso o código esteja sendo executado no lado do servidor, o Meteor se comunicará com a coleção do MongoDB e, assim, poderá ler e escrever os dados diretamente. Por outro lado, se o contexto for o lado do navegador (cliente), o MiniMongo será o responsável por abstrair e processar as funções de acesso, transmissão e modificação dos dados.

No lado do navegador, temos o que chamamos de MiniMongo, uma implementação leve do banco de dados Mongo em JavaScript. Essa implementação é executada diretamente no navegador do cliente e serve como a principal interface de comunicação com o banco de dados persistentes (PLATTS, 2015). O MiniMongo também pode servir como cachê, aumentando, na maioria das vezes, o desempenho geral da aplicação.

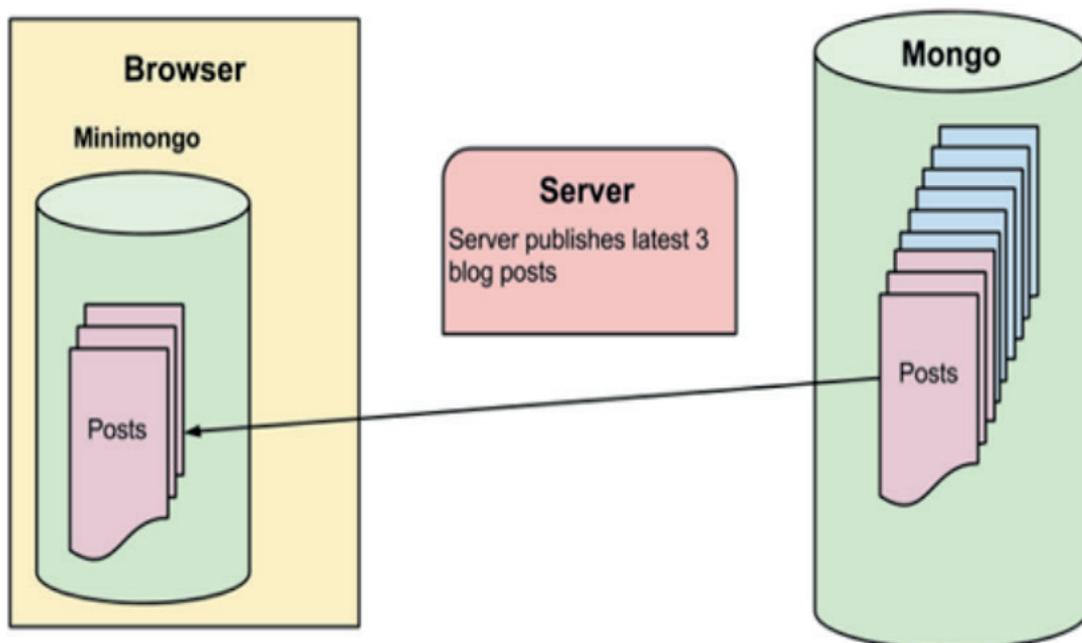


Figura 4: Sincronização de dados feita entre o MongoDB (lado do servidor) e o MiniMongo (lado do cliente). Fonte: Platts, 2015.

Na Figura 4, pode ser visto um exemplo em que o servidor atualiza todos os clientes conectados sobre a publicação de três novos posts. Essa sincronização é automática e pode ser configurada pelo desenvolvedor.

Um ponto importante a se notar é que a coleção do lado do cliente consiste em um subconjunto dos documentos armazenados na coleção Mongo original. Sendo assim, no lado do cliente estão apenas os dados necessários e que sejam visíveis ao usuário atual.

Como os dados do lado do cliente estão na memória do navegador, o acesso é praticamente instantâneo e não são feitas, na maioria dos casos, requisições síncronas adicionais para ler os dados do servidor, visto que a informação já está pré-carregada.

### 3.2.2 Ciclo de vida

Quanto ao ciclo de vida de uma alteração de dados feita pelo usuário na interface gráfica da aplicação, e com o objetivo de melhorar sua experiência, as ações devem ser realizadas o mais rápido possível, com resposta imediata.

Nesse contexto, caso fosse utilizado um modelo tradicional de requisição, o usuário teria que esperar a resposta do servidor para que a interface da aplicação fosse atualizada, e isso tornaria a operação menos fluida. Pensando nisso, foi definida no Meteor uma funcionalidade que é chamada de compensação de latência (STUBAILO, 2015). A Figura 5 detalha essa estratégia.

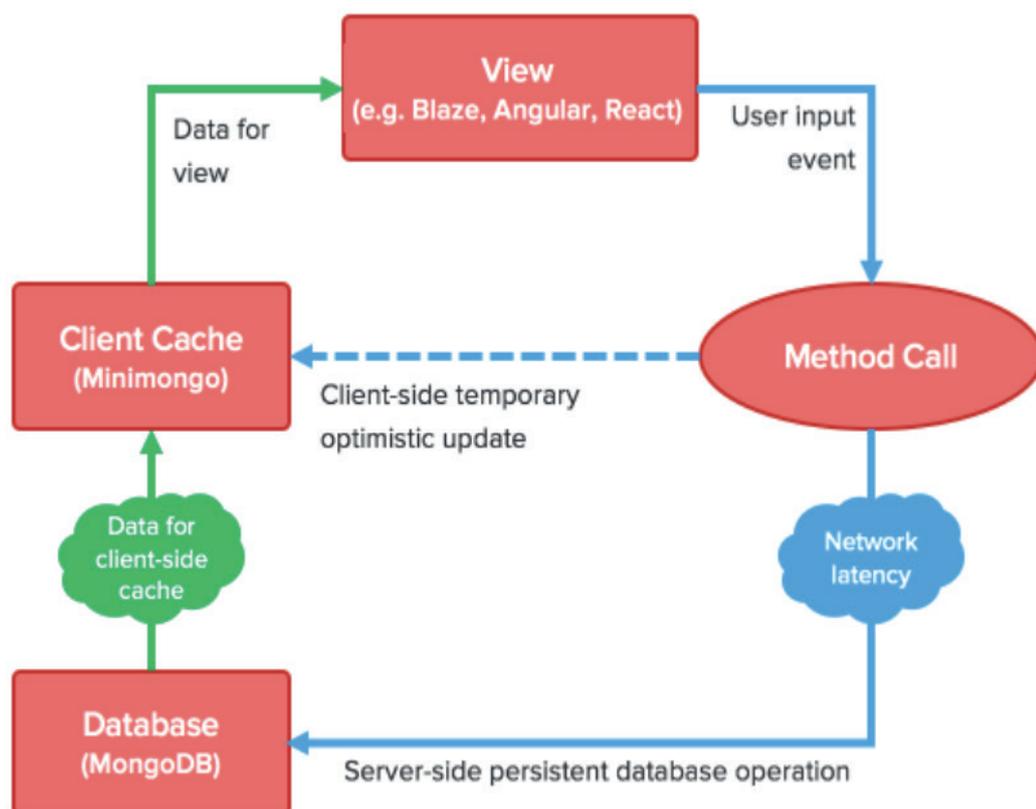


Figura 5: Ciclo de vida de uma requisição no Meteor. Fonte: Stubailo, 2015.

Como o código de verificação dos dados também está disponível no lado do cliente, a chamada ao método atualiza a interface gráfica assumindo que a operação foi realizada com sucesso e, em paralelo, envia uma requisição para executar a ação no lado do servidor de maneira assíncrona.

Na grande maioria das vezes, a requisição do lado do servidor ocorre sem grandes problemas e, como a interface já foi atualizada, não é preciso renderizar a página novamente. Em caso de falha no lado do servidor, a interface gráfica é atualizada e o usuário será informado sobre a falha. Esta técnica também é chamada de interface de usuário otimista, pois se baseia na ideia de que as requisições enviadas pela rede serão executadas com sucesso.

## 4 | CONCLUSÃO

A presente pesquisa apresentou as estratégias utilizadas pelo framework Meteor para a redução de latência e entrega transparente de dados entre o banco de dados e as views renderizadas no navegador da aplicação (lado do cliente). Foram detalhados os componentes da arquitetura e os principais algoritmos utilizados para a sincronização de dados e requisições.

Nesse contexto, a renderização dinâmica de views HTML, aliada à atualização transparente dos dados utilizando coleções, permite uma melhor experiência do usuário, visto que o ciclo de vida da requisição HTTP é drasticamente reduzido.

Como trabalho futuro, pretende-se propor uma estratégia baseada em DDP utilizando o protocolo UDP para possivelmente diminuir a latência entre as requisições.

Em conclusão, a implementação de estratégias de redução de latência em *frameworks* web é cada vez mais comum. O Meteor, por sua vez, utiliza-se de técnicas como o DDP, o MiniMongo e o live query para diminuir a latência percebida pelo usuário ao navegar pela interface gráfica do aplicativo.

## REFERÊNCIAS

BASTOS, Geraldo. **Qual a melhor arquitetura para aplicações iOS? MVVM?** 2017. Disponível em: <<https://medium.com/@GeraldoBastos/parte-2-qual-a-melhor-arquitetura-para-aplica%C3%A7%C3%B5es-ios-mvvm-803f09cbd989>>. Acesso em: 8 maio 2020.

COLEMAN, Tom; GREIF, Sacha. **Discover Meteor: Building real-time Javascript web apps.** [S.l.: s.n.], 2015.

DEBERGALIS, Matt. **Introducing DDP.** 2012. Disponível em: <<https://blog.meteor.com/introducing-ddp-6b40c6aff27d>>. Acesso em: 2 abr. 2020.

DESIGNVELOPER. **Why choose Meteor for your future web and mobile apps?** 2016. Disponível em: <<https://www.designveloper.com/blog/why-choose-meteor-for-your-future-web-and-mobile-apps/>>. Acesso em: 5 maio 2020.

FLANAGAN, David; MATILAINEN, Pasi. **JavaScript**. [S.l.]: Anaya Multimedia, 2007.

GOOGLE. **AngularJS: Superheroic JavaScript MVW framework**. 2019. Disponível em: <<https://angularjs.org/>>. Acesso em: 2 maio 2020.

MONGODB INC. **Meteor: Build iOS and Android apps that are a delight to use**. 2015. Disponível em: <<https://www.mongodb.com/blog/post/meteor-build-ios-and-android-apps-are-delight-use>>. Acesso em: 1 maio 2020.

OPENJS FOUNDATION. **Node.js**. 2020. Disponível em: <<https://nodejs.org/en/>>. Acesso em: 1 maio 2020.

PLATTS, Matthew. **Meteor tutorial: bring your ideas to life with Meteor**. Melbourne, Sydney: Web Tesmpest, 2015.

STRACK, Isaac. **Getting started with Meteor.js JavaScript framework**. [S.l.]: Packt Publishing Ltd, 2015a.

\_\_\_\_\_. **Why Meteor rocks!** 2015b. Disponível em: <<https://hub.packtpub.com/why-meteor-rocks/>>. Acesso em: 5 abr. 2020.

STUBAILO, Sashko. **Optimistic UI with Meteor**. 2015. Disponível em: <<https://blog.meteor.com/optimistic-ui-with-meteor-67b5a78c3fcf>>. Acesso em: 9 maio 2020.

TILKOV, Stefan; VINOSKI, Steve. Node.js: Using JavaScript to build high-performance network programs. **IEEE Internet Computing**, v. 14, n. 6, p. 80-83, 2010.

TIMMS, Simon. **Mastering JavaScript design patterns**. [S.l.]: Packt Publishing Ltd, 2014.

## ÍNDICE REMISSIVO

### A

Aeromodelo 123, 128

Algoritmo 19, 20, 48, 50, 53, 54, 55, 57, 58, 90, 138, 172, 173, 174, 180, 182

AngularJS 37, 38, 39, 43, 47

Aplicativos 38, 60, 61, 62, 63, 64, 65, 66, 67, 99, 101, 146, 174

Aprendizado de máquina 48, 50, 58

Aprendizagem 26, 27, 31, 33, 34, 35, 58, 70, 71, 72, 75, 77, 78, 79, 80, 81, 84, 87, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 137, 197, 198, 199, 200, 206

Arduino 28, 29, 30, 34, 124, 130, 131, 197, 198, 199, 201, 202, 203, 205, 206

Arquitetura 4, 6, 7, 11, 36, 37, 39, 40, 41, 42, 43, 46, 50, 51, 52, 60, 65, 66, 67, 76, 93, 97, 100, 130, 156, 184, 186, 187, 190, 191, 192, 194

Ataques cibernéticos 172, 174, 176, 180

Automação 1, 3, 5, 9, 13, 26, 28, 30, 33, 90

### B

Banco de dados 4, 8, 10, 20, 36, 37, 40, 41, 42, 43, 44, 46, 51, 52, 54, 70, 74, 93, 156, 167, 174, 175, 176, 207

Bateria 76, 121, 124, 125, 126, 127, 129, 130

Blockchain 183, 184, 185, 186, 187, 188, 190, 191, 194, 195, 196

BPMN 4, 9, 13, 14, 151, 152, 153, 154, 155, 156, 158, 160, 161, 164, 165

### C

Cálculo espectral 107, 109, 116

Ciência da informação 60, 61, 68, 207

Circuitos elétricos 26, 28, 30, 33, 203

Competência 78, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93

Computação 1, 35, 36, 66, 70, 71, 72, 76, 81, 82, 83, 84, 85, 87, 88, 89, 90, 91, 92, 93, 94, 103, 105, 121, 137, 201, 207

Contagem de pedestre 133, 138, 139

### D

Digital 1, 2, 60, 61, 62, 68, 69, 80, 104, 123, 129, 133, 134, 143, 144, 146, 173, 182, 187, 190, 193

Diretrizes curriculares 82, 83, 94

Disciplina 72, 75, 76, 77, 80, 82, 83, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 102, 103, 104, 160

Dispersão criptográfica 172, 173

Drone 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132

## E

Educação 3, 4, 5, 6, 7, 14, 27, 28, 33, 34, 35, 82, 83, 96, 104, 105, 107, 173, 183, 194, 195, 198, 206, 207

Elicitação de requisitos 151, 152, 154, 156, 164, 165

Engenharia de software 70, 71, 72, 80, 81, 165

Ensino 5, 7, 26, 27, 28, 29, 34, 35, 70, 71, 72, 74, 75, 77, 78, 80, 84, 85, 87, 88, 91, 93, 94, 95, 96, 97, 98, 99, 104, 105, 106, 143, 160, 197, 198, 199, 200, 206

## F

Formação 13, 25, 34, 72, 82, 83, 84, 86, 87, 91, 92, 94, 105, 119, 200

Framework 5, 8, 36, 37, 40, 41, 43, 46, 47, 156, 182

Front-end 37, 39, 40, 43

Full-stack 36, 37, 40, 43

## I

Imagens médicas 15, 16, 17, 19, 25

Informação 1, 2, 3, 4, 5, 7, 8, 14, 15, 16, 20, 24, 25, 26, 27, 36, 45, 48, 49, 52, 59, 60, 61, 62, 63, 66, 67, 68, 69, 70, 75, 76, 77, 82, 84, 85, 89, 91, 95, 107, 120, 133, 134, 137, 143, 144, 146, 149, 150, 151, 160, 164, 166, 167, 168, 172, 183, 184, 185, 186, 188, 189, 190, 197, 204, 207

Inteligência artificial 48, 49, 58, 62

Internet 33, 47, 51, 62, 69, 75, 100, 101, 103, 144, 149, 150, 172, 182, 185, 195, 196

Islandtest 70, 71, 72, 74, 75, 76, 77, 78, 79, 80

## J

Java 38, 51, 87, 156, 178, 191

Javascript 37, 38, 40, 41, 43, 44, 47, 70, 71, 74, 75, 173, 177, 181, 182, 190

Jogos 29, 31, 32, 33, 70, 71, 72, 79, 80, 81, 206

Jogos educativos 70, 72, 80

## L

Laboratório 7, 29, 31, 33, 168

Linguagem R 107, 110

## M

Manutenção 2, 5, 6, 10, 43, 63, 72, 109, 158

Message-Digest Algorithm 172, 173, 181

MeteorJS 36, 37, 40

ML-SAI 95, 96, 97, 99, 101, 102, 104, 105  
Mobile learning 96, 98, 104  
Modelo pedagógico 95, 96, 97, 99, 100, 101, 102, 104, 105  
MongoDB 37, 40, 43, 44

## **N**

node.js 37, 41, 47

## **O**

Ontologia 1, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13

## **P**

Programação 19, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 37, 38, 39, 51, 82, 85, 86, 87, 88, 89, 90, 91, 92, 93, 95, 96, 99, 102, 103, 104, 105, 124, 137, 168, 173, 201, 203, 205, 207  
Protótipo 54, 129, 203, 205  
Python 52, 87, 88, 137, 173, 181

## **Q**

Quadricóptero 126, 129, 130  
QuantumGIS 107, 108

## **R**

Raciocínio lógico 30, 33, 34  
Rastreamento 51, 133, 134, 135, 137, 138, 140, 141  
Reator nuclear 197, 199, 200, 202, 204, 205  
Redes sociais 99, 143, 144, 145, 146, 147, 148, 149, 150, 192  
Rede YOLO 135, 136, 137, 140  
Requisitos 74, 77, 78, 79, 81, 101, 120, 122, 126, 134, 151, 152, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165  
Robótica 26, 27, 28, 29, 30, 32, 33, 34, 35, 90

## **S**

Sala de aula invertida 95, 96, 98, 99, 100, 104, 105, 106  
Scratch 26, 28, 29, 30, 31, 32, 33  
Segurança 42, 48, 76, 107, 120, 122, 126, 134, 143, 144, 145, 146, 149, 150, 172, 173, 174, 181, 185, 186, 187, 190, 192, 194  
Simulador 197, 199, 200, 201, 202, 203, 204, 205, 206  
Sincronização de dados 36, 37, 44, 46

Sistema de informação ambiental 166

Sistemas de informação 1, 15, 26, 36, 48, 49, 59, 60, 69, 70, 75, 76, 77, 82, 84, 91, 95, 107, 120, 133, 143, 151, 166, 172, 183, 197, 207

Sustentabilidade 1, 2, 13

## **T**

Técnica REMO 151, 152, 154, 155, 156, 158, 160, 161, 162, 163, 164, 165

Tecnologia 3, 4, 5, 7, 1, 2, 14, 27, 28, 29, 33, 49, 51, 60, 61, 62, 63, 66, 67, 68, 69, 77, 82, 98, 99, 100, 105, 107, 123, 129, 132, 137, 150, 151, 152, 162, 183, 184, 185, 186, 187, 188, 189, 190, 192, 194, 195, 196, 197, 199, 200, 201, 204, 207

Tecnologia da informação 2, 14, 60, 61, 62, 63, 68, 184, 188, 204, 207

Testes de software 70, 71, 72, 74, 75, 76, 77, 79, 80

## **V**

VANT 121, 126, 128, 130, 131

Visão computacional 15, 24, 137

## **W**

Web de dados 60, 61

## **X**

XPDL 151, 152, 153, 154, 156, 157, 158, 161, 164, 165

# Sistemas de Informação e Aplicações Computacionais

[www.atenaeditora.com.br](http://www.atenaeditora.com.br) 

[contato@atenaeditora.com.br](mailto:contato@atenaeditora.com.br) 

[@atenaeditora](https://www.instagram.com/atenaeditora) 

[www.facebook.com/atenaeditora.com.br](https://www.facebook.com/atenaeditora.com.br) 

Atena  
Editora

Ano 2020

# Sistemas de Informação e Aplicações Computacionais

[www.atenaeditora.com.br](http://www.atenaeditora.com.br) 

[contato@atenaeditora.com.br](mailto:contato@atenaeditora.com.br) 

[@atenaeditora](https://www.instagram.com/atenaeditora) 

[www.facebook.com/atenaeditora.com.br](https://www.facebook.com/atenaeditora.com.br) 

Atena  
Editora

Ano 2020