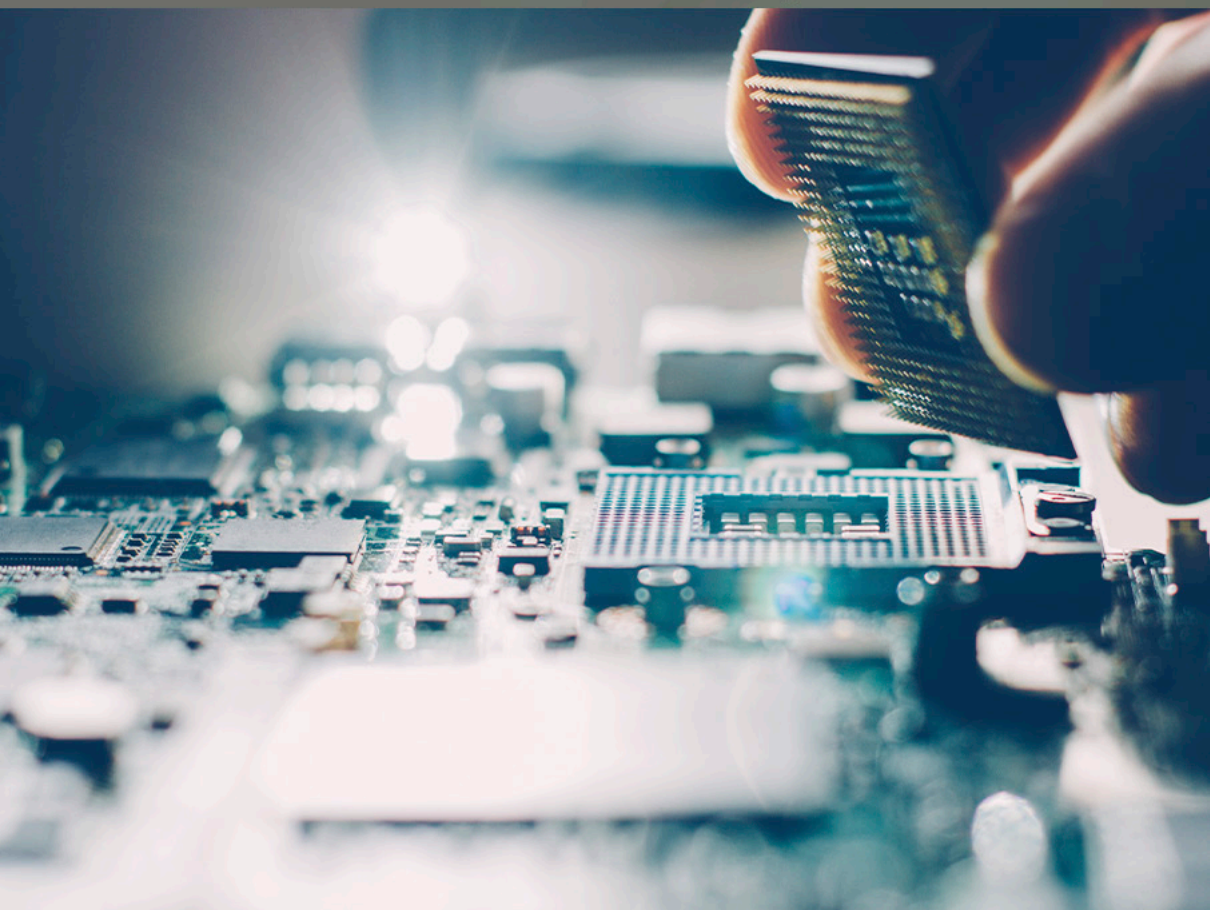


COLEÇÃO

# DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO 3

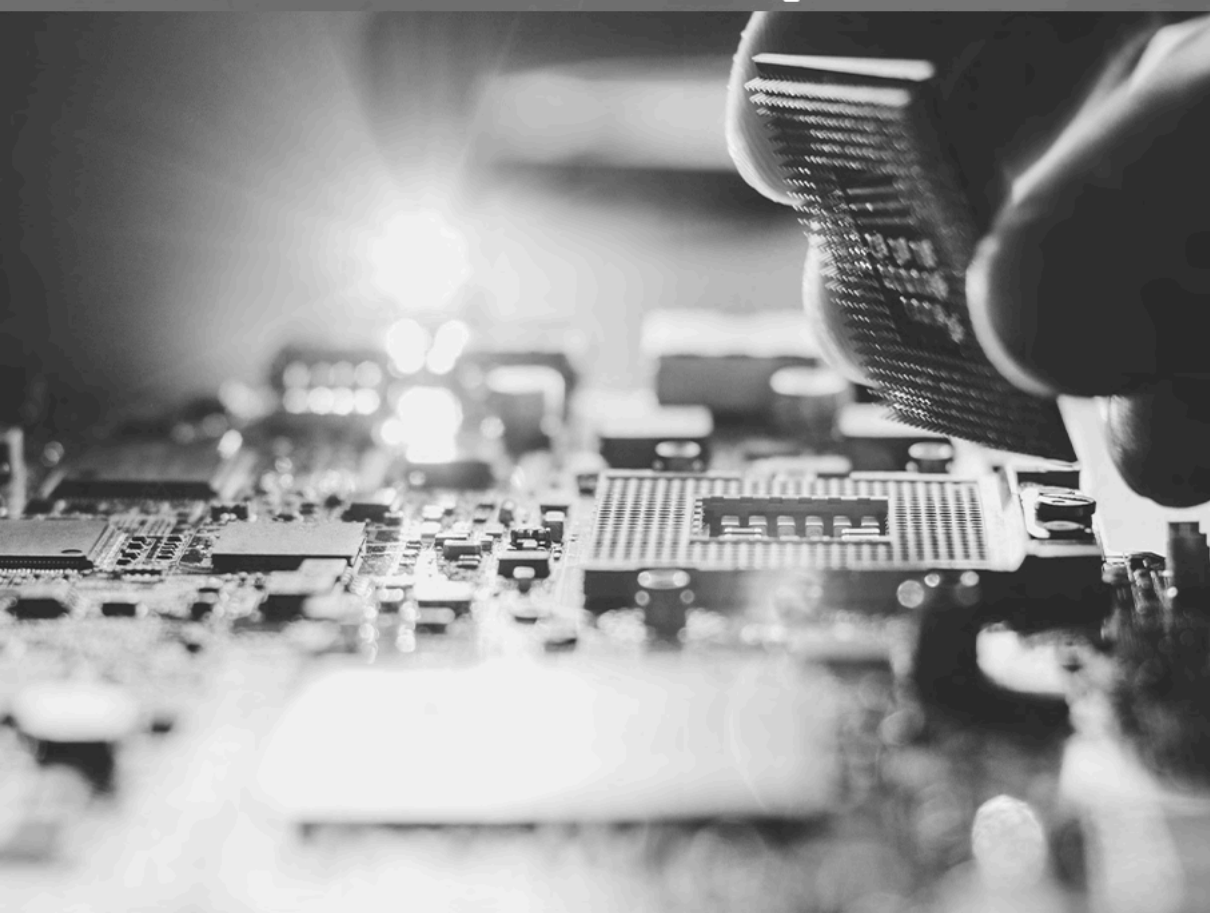


LILIAN COELHO DE FREITAS  
(ORGANIZADORA)

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

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

<b>CAPÍTULO 11</b> .....	<b>119</b>
ESTUDO SOBRE AUTOMATIZAÇÃO DE EQUIVALÊNCIA DE FUNÇÕES	
Lucas Fernando Frighetto Fábio Hernandez	
 <a href="https://doi.org/10.22533/at.ed.19221291111">https://doi.org/10.22533/at.ed.19221291111</a>	
<b>CAPÍTULO 12</b> .....	<b>142</b>
ESTUDO SOBRE O CONTROLE REMOTO DE DISPOSITIVOS MICROCONTROLADOS UTILIZANDO DISPOSITIVOS MÓVEIS	
João Vítor Fernandes Dias Fermín Alfredo Tang Montané	
 <a href="https://doi.org/10.22533/at.ed.19221291112">https://doi.org/10.22533/at.ed.19221291112</a>	
<b>CAPÍTULO 13</b> .....	<b>163</b>
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	
 <a href="https://doi.org/10.22533/at.ed.19221291113">https://doi.org/10.22533/at.ed.19221291113</a>	
<b>CAPÍTULO 14</b> .....	<b>174</b>
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	
 <a href="https://doi.org/10.22533/at.ed.19221291114">https://doi.org/10.22533/at.ed.19221291114</a>	
<b>CAPÍTULO 15</b> .....	<b>186</b>
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	
 <a href="https://doi.org/10.22533/at.ed.19221291115">https://doi.org/10.22533/at.ed.19221291115</a>	
<b>CAPÍTULO 16</b> .....	<b>199</b>
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	
 <a href="https://doi.org/10.22533/at.ed.19221291116">https://doi.org/10.22533/at.ed.19221291116</a>	

<b>CAPÍTULO 17</b> .....	<b>224</b>
FINITE ELEMENT METHOD APPLIED TO MECHANICAL ANALYSIS OF AERONAUTICAL RIBS IN CARBON FIBER AND 7075 ALUMINUM ALLOY	
Alex Fernandes de Souza	
 <a href="https://doi.org/10.22533/at.ed.19221291117">https://doi.org/10.22533/at.ed.19221291117</a>	
<b>CAPÍTULO 18</b> .....	<b>236</b>
MÉTODO PARA CALCULAR A ÁREA DE SUPERFICIAL DE RAÍZES POR PROCESSAMENTO DIGITAL DE IMAGENS	
Marcio Hosoya Name	
 <a href="https://doi.org/10.22533/at.ed.19221291118">https://doi.org/10.22533/at.ed.19221291118</a>	
<b>CAPÍTULO 19</b> .....	<b>244</b>
LOCAL MESHFREE METHOD OPTIMIZATION WITH GENETICALGORITHMS	
Wilber Vélez	
Flávio Mendonça	
Artur Portela	
 <a href="https://doi.org/10.22533/at.ed.19221291119">https://doi.org/10.22533/at.ed.19221291119</a>	
<b>CAPÍTULO 20</b> .....	<b>258</b>
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	
 <a href="https://doi.org/10.22533/at.ed.19221291120">https://doi.org/10.22533/at.ed.19221291120</a>	
<b>SOBRE A ORGANIZADORA</b> .....	<b>268</b>
<b>ÍNDICE REMISSIVO</b> .....	<b>269</b>

# CAPÍTULO 12

## ESTUDO SOBRE O CONTROLE REMOTO DE DISPOSITIVOS MICROCONTROLADOS UTILIZANDO DISPOSITIVOS MÓVEIS

Data de aceite: 01/11/2021

Data de submissão: 04/08/2021

### João Vítor Fernandes Dias

UNIVERSIDADE ESTADUAL DO NORTE  
FLUMINENSE DARCY RIBEIRO  
Campos dos Goytacazes – Rio de Janeiro  
<http://lattes.cnpq.br/2335804614447803>  
<https://orcid.org/0000-0002-8156-9551>

### Fermín Alfredo Tang Montané

UNIVERSIDADE ESTADUAL DO NORTE  
FLUMINENSE DARCY RIBEIRO  
Campos dos Goytacazes – Rio de Janeiro  
<http://lattes.cnpq.br/9974379145983363>  
<https://orcid.org/0000-0002-2637-6562>

**RESUMO:** A Internet das Coisas (*Internet of Things* – IoT) tem como essência a interligação de objetos físicos mediante tecnologias de conexão em rede. Este projeto tem como objetivo a montagem de um braço robótico e a programação de um aplicativo Android que permita controlá-lo de forma remota. As metodologias utilizadas foram a programação na plataforma Arduino, para o acionamento do braço robótico, e no ambiente Visual Studio Code, para o desenvolvimento do aplicativo; montagem da estrutura e componentes do braço robótico; e a integração do aplicativo com o código Arduino que o movimentava. Como resultado foram pesquisadas diversas tecnologias de comunicação remota, sendo escolhidas para o projeto as tecnologias wi-fi e Bluetooth. Testes

foram realizados usando os microcontroladores NodeMCU com wi-fi integrado e o módulo HC-05 com conexão Bluetooth. Foram pesquisadas linguagens de programação com foco em desenvolvimento de apps, sendo escolhido o JavaScript em conjunto com o framework React Native. Um app para smartphones Android foi desenvolvido para comunicação Bluetooth com diversas funcionalidades, podendo então se conectar ao módulo HC-05 e enviar os dados. Assim, o aplicativo é capaz de se conectar através de uma tecnologia de comunicação remota à um dispositivo microcontrolado capaz de movimentar o braço robótico. Posteriormente, passou-se a testar a conexão wi-fi do microcontrolador NodeMCU Amica com resultados satisfatórios. Como conclusão podemos observar o aprendizado quanto à diferentes tecnologias de comunicação. Em relação às linguagens de programação utilizáveis para o desenvolvimento de aplicativos, foram escolhidos o JavaScript e o React Native, com os quais foi desenvolvido um aplicativo capaz de controlar o braço robótico através do HC-05 integrado ao Arduino, e o desenvolvimento de um servidor web com o uso do microcontrolador NodeMCU para envio de dados através do wi-fi e a montagem de um braço robótico.

**PALAVRAS-CHAVE:** Arduino, Bluetooth, JavaScript, Microcontrolador, Wi-fi

### STUDY ABOUT REMOTE CONTROLLING MICROCONTROLLED DEVICES USING MOBILE DEVICES

**ABSTRACT:** The Internet of Things (IoT) has as

essence the interlinking between physical objects by technologies of network connection. This project has as its main objective the assembly of a robotic arm and the development of an Android app that allows its remote control. The used methodologies were the programming in the Arduino platform, to activate the robotic arm, and in the Visual Studio Code environment, for the development of an App; assembly of the structure and the components of the robotic arm; and the integration of the Android app with the Arduino code that movements the robotic arm. As a result, a huge variety of technologies of remote control have been researched, being chosen for the project the technologies wi-fi and Bluetooth. Test were realized using the microcontrollers NodeMCU with embedded wi-fi and the HC-05 module with Bluetooth connection. Programming languages focused on app Development were researched, being chosen JavaScript together with the React Native framework. An app for Android smartphones was developed to communicate via Bluetooth with a variety of functionalities, being able to connect to the HC-05 module and send date. The app, is then, capable of connecting through a remote controlling technology to a microcontrolled device capable of moving a robotic arm. Subsequently the wi-fi from the NodeMCU Amica was tested, with satisfying results. As a conclusion the learning of different remote controlling technologies can be observed. Regarding the applicable programming languages for the app development, JavaScript and the React Native framework were chosen, with which was developed an app capable of controlling a robotic arm through the HC-05 together with the Arduino, and the development of a web server with the use of the NodeMCU microcontroller to send date through wi-fi and the assembly of a robotic arm.

**KEYWORDS:** Arduino, Bluetooth, JavaScript, Microcontroller, Wi-fi

## 1 | ETAPAS PROPOSTAS NO PLANO DE TRABALHO

O plano de trabalho contempla as seguintes propostas:

- a. Estudo sobre diferentes tecnologias de comunicação remota para controle de dispositivos microcontrolados, entre elas: Bluetooth, Radiofrequência, WiFi, etc. Pesquisa sobre módulos eletrônicos disponíveis no mercado e documentação do estudo.
- b. Estudo sobre alternativas de desenvolvimento de aplicativos Android visando o desenvolvimento de interfaces de controle e monitoramento para dispositivos microcontrolados. Documentação do estudo.
- c. Estudo, desenvolvimento e implementação do segundo protótipo de braço robótico. Escolha da plataforma microcontrolada. Documentação.
- d. Desenvolvimento da interface de controle e monitoramento do braço robótico. Documentação.
- e. Realização de testes de avaliação e desempenho da interface e do braço robótico.
- f. Elaboração de relatório técnico.

Quanto ao item a foi decidido utilizar as tecnologias de Bluetooth e wi-fi visto que

são as únicas amplamente utilizadas nos Smartphones. No item *b* foram vistas diferentes linguagens de programação apropriadas para o desenvolvimento de apps, e dentre elas foi escolhida a linguagem React Native devido à possibilidade de rápida checagem de erros no código e também poder utilizar o aplicativo (app) para dispositivos iOS futuramente. O item *c* foi parcialmente concluído. As plataformas microcontroladas escolhidas para teste foram NodeMCU Amica/Lolin e se mostraram viáveis. Porém após o teste com essas placas, o módulo Bluetooth acabou gerando resultados imprevistos que ainda não foram solucionados, suspeita-se de uma falha no hardware e foi encomendado um novo módulo de reposição. A interface desenvolvida para o item *d* se mostrou eficiente, porém apresenta alguns bugs que deverão ser resolvidos no futuro. Os testes realizados para o item *e* mostram que, apesar de funcional, diversos pequenos detalhes podem ser aprimorados para resultar em uma melhor manipulação do braço mecânico, principalmente em relação ao app.

## 2 | INTRODUÇÃO

A Internet das Coisas (Internet of Things – IoT) tem como essência a interligação de diferentes tecnologias de rede agregadas a objetos, coisas, como smartphones, sensores pessoais, braços robóticos na automação, dentre outros. Essa interligação se torna uma única malha de dispositivos conectados. Mesmo que inicialmente tivessem propósitos únicos, ao se interligar passam a ter uma nova gama mais extensa de funcionalidades que se complementam [Ste18], [San14].

Como citado acima, o uso de braços robóticos está presente na realidade do IOT na escala industrial relacionado a automação, porém não está limitado a ela. Seu uso se expande a diversas funções [aKa18], tais como usar um braço mecânico leve e de alta velocidade para jogar badminton, utilizar um braço mecânico para a escrita [Lee20] ou até mesmo utilizado nas sondas espaciais Perseverance, Spirit, Opportunity e Curiosity para carregar câmeras [Dun171], [Dun17].

O controle desses braços robóticos pode ser feito de diversas formas: controle automático e repetitivo como em uma linha de produção ou com ações baseadas em respostas a sensores, ou também controlados remotamente por uma interface. Com o aumento da popularidade de smartphones e seus aplicativos [Biø19], a capacidade de controlar dispositivos à distância através dos dispositivos mobile se torna mais almejado. Para o desenvolvimento de uma interface que permita essa função, busca-se um bom aproveitamento de código para as diversas plataformas disponíveis atualmente, com destaque para os dispositivos iOS e Android, sem que suas funcionalidades sejam consideravelmente comprometidas, essa capacidade se vê presente na linguagem React Native o que o torna uma alternativa viável [Jaw18].

### 3 | OBJETIVOS

Este projeto tem como objetivo o estudo de tecnologias de comunicação remota, assim como o desenvolvimento de um aplicativo para smartphones Android com interface de controle para um novo tipo de braço robótico, com cinco graus de liberdade e motores de maior potência. Dessa forma, procura-se tornar o projeto do novo bolsista independente do que foi desenvolvido pela bolsista anterior. Procura-se acrescentar novos conhecimentos à experiência anterior.

### 4 | METODOLOGIA

Para o desenvolvimento do plano de trabalho foi seguida a seguinte metodologia:

#### 4.1 Pesquisas e minicursos

Pesquisa de blogs explicativos sobre módulos de comunicação remota [Koy21] e servomotores [Wil21], palestras sobre Bluetooth [Lee18]. Pesquisa de diversos artigos científicos na plataforma Periódicos Capes [Cap21] Participação de minicursos gratuitos de React Native ministrados pelo canal Developer Plus [Plu18], [Plu19] e a instituição educacional RocketSeat [Sea21] durante a Next Level Week #5. Participação de minicursos online voltados ao aprendizado da pesquisa científica em portais de periódicos ministrado por funcionários relacionados à plataforma Periódicos Capes [Tre21].

#### 4.2 Programação no Arduino IDE

Desenvolvimento no ambiente Arduino IDE, que é um ambiente de desenvolvimento integrado (em inglês: *Integrated Development Environment* – IDE) que disponibiliza um editor de texto para a escrita do código, um console de texto, barra de ferramentas com diversas funções e permite a conexão com o Arduino UNO para o envio do programa desenvolvido e se comunicar com o dispositivo [Ard21]. É o software comumente utilizado para se desenvolver projetos utilizando microcontroladores Arduino, porém também pode ser utilizado para o desenvolvimento em outros microcontroladores. A Figura 1, ilustra a interface do Arduino IDE.



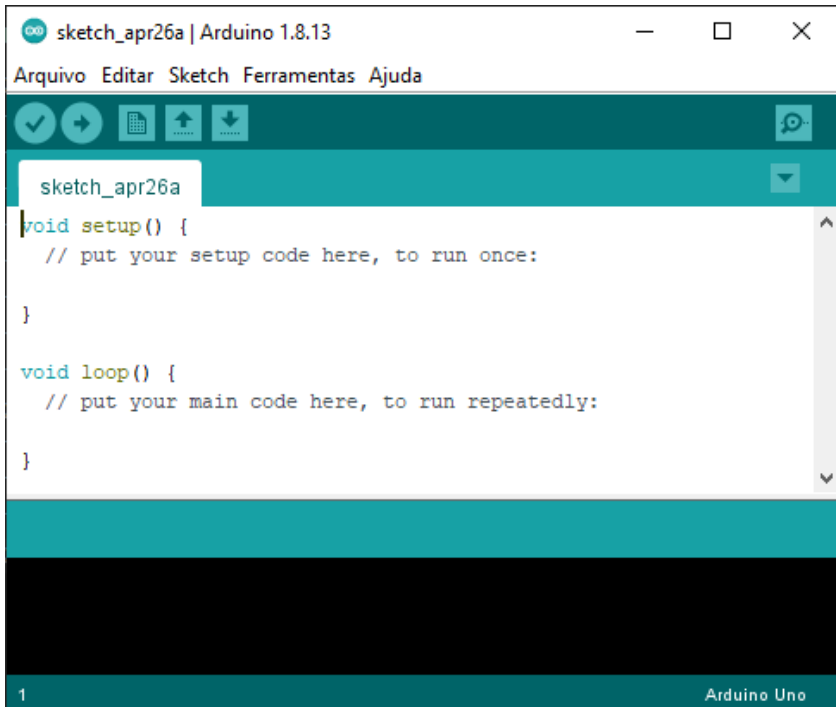


Figura 1: Tela inicial do Arduino IDE 1.8.13

### 4.3 Programação de aplicativos no Visual Studio Code

Desenvolvimento no ambiente Visual Studio Code (ou também “VS Code”), que é um editor de código simplificado com suporte para operações de desenvolvimento como debug e controle de versões [FAQ21]. Possui também a característica IntelliSense que permite o complemento automático do código enquanto está sendo escrito o que reduz a chance de erros provenientes da datilografia (geralmente chamado de “typo” na área de programação) e também torna mais eficiente e rápido o desenvolvimento do código [Vis21]. Esse foi o ambiente utilizado para o desenvolvimento de aplicativos mobile e que com o uso de extensões também permite a substituição do Arduino IDE, se mostrando uma ferramenta ainda mais completa. Este ambiente é ilustrado na Figura 2.

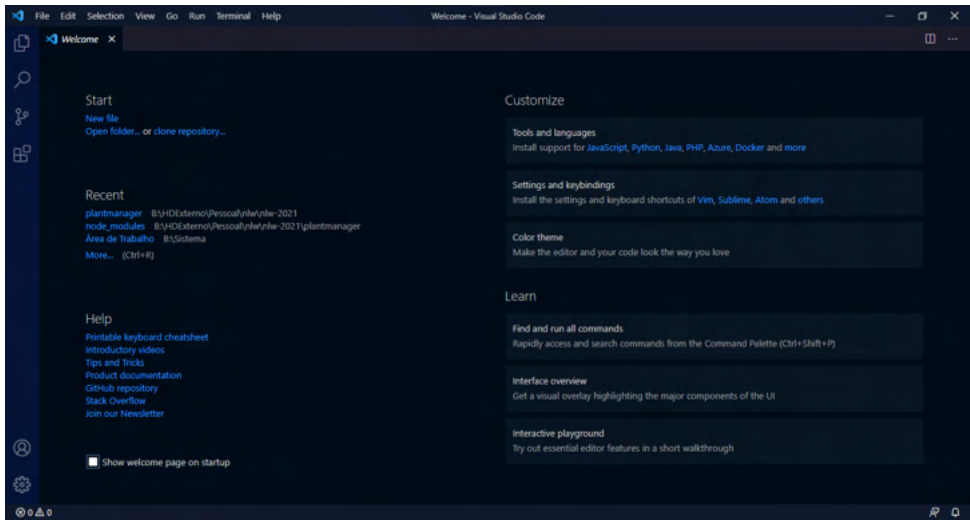


Figura 2: Visual Studio Code 1.55.2.

## 4.4 Montagem do braço robótico

A montagem da estrutura e componentes do novo braço robótico é uma tarefa manual, que exige atenção aos detalhes, e o uso de ferramentas. Foram utilizados uma chave Philips e o manual disponibilizado pelo fornecedor das peças do braço robótico, para montar o novo braço robótico com 5 graus de liberdade, sendo este novo braço feito em acrílico, diferentemente da primeira versão feita em material MDF.

## 4.5 Junção dos códigos com os dispositivos

Após o reaproveitamento dos códigos desenvolvidos anteriormente no Arduino IDE, e o aprendizado sobre programação de aplicativos seguido de seu desenvolvimento, ambos os códigos foram integrados e permitiram a transmissão de dados entre um dispositivo Android e um módulo Arduino através de conexão Bluetooth. Posteriormente com a montagem do braço robótico, a transmissão de dados, já funcional, foi utilizada para movimentar o braço robótico.

# 5 | RESULTADOS E DISCUSSÃO

Como resultado vê-se de forma sucinta a montagem do braço robótico e do aprendizado sobre tecnologias de comunicação remota adquirido para o desenvolvimento do app.

## 5.1 Seleção das tecnologias de comunicação remota

Nesta pesquisa foram encontradas diversas tecnologias de comunicação remota, muitas delas utilizadas no contexto do IOT. Apresentam-se as descrições dessas tecnologias

junto a um parecer quanto a sua adequação ao atual projeto.

RFid (*Radio-Frequency Identification* – Identificação por Radiofrequência) – Não apropriado: Ações limitadas ao que for programado em cartões de RFid, não apropriado devido a necessidade da alta taxa de repetição e controle preciso dos servomotores.

Protocolo CAN – Viável, porém não apropriado: Permite envio de até 11 bits ou 29 bits e requer um dispositivo de envio e outro de recebimento, isso o torna viável, ou seja, seria possível desenvolver o controle do braço robótico utilizando essa tecnologia, porém, ao inserir módulos MCP2515 [Koy181] como intermediários aumentaria desnecessariamente a infraestrutura em comparação com as outras tecnologias escolhidas, além disso, não pode ser controlado diretamente por um smartphone que é um dos objetivos da pesquisa, o que faz com que não seja apropriado.

ESP32 LoRaWan – Viável, porém não apropriado: Depende de outro LoraWan [Koy18], e o foco da pesquisa é o controle do braço robótico através do Smartphone, o que faz com que entre no mesmo critério eliminatório que o do Protocolo CAN.

ZigBee – Viável, porém não apropriado: O ZigBee é um protocolo de comunicação feito para se adequar aos requisitos de dispositivos embarcados. Provê baixo consumo de energia e suporta grande número de dispositivos através de longas distâncias com várias topologias diferentes [Ban19]. Dependendo do avanço desse projeto, pode vir a ser apropriado o uso, desse protocolo, porém atualmente não se mostra adequado às necessidades.

6LoWPAN – Apropriado (mas não escolhido): Segundo [Nik19] “o ‘IPv6 em rede sem fio pessoal de baixa energia’ (IPv6 over *Low Power Wireless Personal Area Network* - 6LoWPAN) é uma adaptação da subcamada do IPv6 e provê conectividade por IP em redes de baixa energia e com perda de dados; o IETF padronizou essa subcamada. Hoje, 6LoWPAN é a tecnologia chave para vários modelos de rede no IoT assim como automação residencial, controle de sistemas industriais e cidades inteligentes.” Mesmo havendo a possibilidade do uso do módulo L-Tek 6LoWPAN Arduino Shield 900MHz ou do AxAvior 6LoWPAN Module, por enquanto optou-se por não utilizar essa tecnologia por causa da necessidade da obtenção dos módulos. Porém pode ser alvo de maiores pesquisas posteriormente, assim como o BLE (*Bluetooth Low Energy*).

Wi-fi (*Wireless Fidelity*) – Apropriado: Também chamado de WLAN (*Wireless Local Area Network* – Rede Sem Fio de Área Local) é uma tecnologia presente na maioria dos Smartphones atuais, geralmente utilizado para se conectar à internet. Apresenta diversos módulos para o uso com o Arduino. Estão disponíveis vários microcontroladores, como o ESP32 e o NodeMCU ESP8266, que já vêm com o módulo wi-fi embutido. Como o NodeMCU ESP8266 está disponível para uso sem a necessidade de aquisição, ele foi escolhido para ser testado.

Bluetooth (Classic) – Apropriado: O Bluetooth, também chamado de WPAN (*Wireless Personal Area Network* – Rede Sem Fio de Área Pessoal), assim como o wi-fi, também é uma tecnologia presente na maioria dos Smartphones atuais, geralmente utilizado para se

conectar dispositivos sem fio, principalmente fones de ouvido, teclados e mouses. Apresenta diversos módulos para o uso com o Arduino, podendo ser citados os módulos HC-05 (utilizado nesse projeto) e o HC-06. Essa tecnologia também está disponível no módulo ESP32 citado anteriormente que embora não disponível, pode ser adquirido para projetos futuros.

Conclusão: As tecnologias apropriadas e escolhidas para o projeto atualmente são wi-fi e Bluetooth. Foram testados os microcontroladores NodeMCU Amica com ESP8266 e NodeMCU Lolin com ESP8266 para a conexão wi-fi e foi utilizado o módulo HC-05 para a conexão Bluetooth com o Arduino e também com ambos os microcontroladores Amica e Lolin.

## 5.2 Seleção dos microcontroladores

Quanto às plataformas microcontroladas foram pesquisadas diversas opções como:

Placas que têm o microchip ESP8266: ESP01, ESP12E, ESP12F, ESP201, NodeMCU-ESP12 e CP2102 e placas que têm o chip ESP32: ESP32-WROVER, ESP32-WROOM-32U, ESP32S-CP2102 e NodeMCU V3 WIFI 802.11.

De início optou-se por manter o desenvolvimento com o Arduino, visto que a quantidade de falhas poderia ser reduzida devido à familiaridade, e por isso o desenvolvimento pode ser mais dinâmico. Assim que o desenvolvimento fosse concluído, novos testes poderiam ser realizados com as outras placas para analisar se vale ou não a pena a mudança. As placas disponíveis para testes no laboratório e que foram definidas como viáveis para o projeto através de pesquisa temos o NodeMCU Amica e NodeMCU Lolin ambos com o microchip ESP8266 com acesso à conexão wi-fi.

## 5.3 Códigos no Arduino IDE

Com o objetivo final de controlar o braço robótico através de tecnologias de comunicação remota, foram desenvolvidos programas no Arduino IDE para se alcançar este resultado. Nas subseções seguintes apresentam-se as ferramentas utilizadas assim como as etapas realizadas.

### 5.3.1 *Monitor e Plotter Serial*

A forma utilizada para a análise dos valores recebidos pontualmente é o uso do monitor Serial. A figura 3 ilustra o uso do monitor serial para mostrar o valor dos ângulos de cada um dos servomotores separadamente.

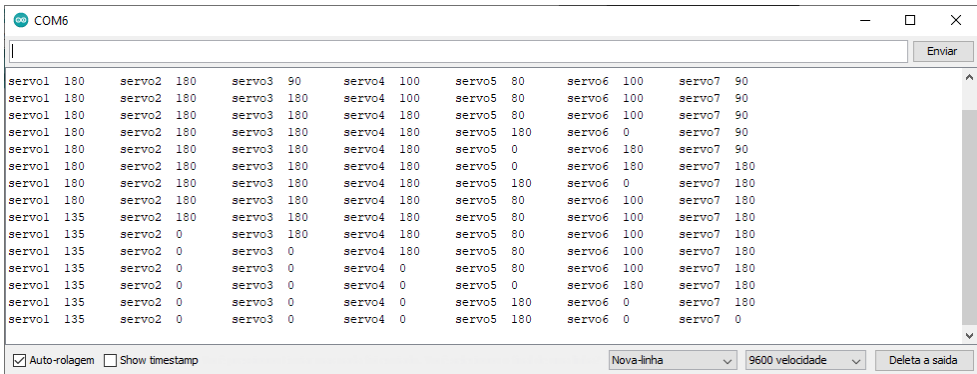


Figura 3: Demonstração do Monitor Serial.

De acordo com as seguintes funções:

```
void printServo (int pos, char servo){
    Serial.print("servo");
    Serial.print(servo);    tab();
    Serial.print(pos);    tab();
}
```

Código 1: Função genérica para imprimir o valor de qualquer servo.

O código 1 mostra a função genérica *printServo ()* que imprime a informação de angulação e identificação do servo. Para isso, imprime o texto “servo”, seguido de um caractere referente a qual servo terá sua angulação mostrada, logo após há uma função *tab ()* que apenas imprime uma tabulação para deixar devidamente organizado o texto na saída, e enfim imprime um valor inteiro correspondente à posição do servo (variável “*pos*”) com mais uma tabulação.

```
void servoMonitor () {
    printServo (ang1, '1');
    printServo (ang2, '2');
    printServo (ang3, '3');
    printServo (ang4, '4');
    printServo (ang5, '5');
    printServo (ang6, '6');
    printServo (ang7, '7');
    line ();
}
```

Código 2: Função que chama a função “printServo ()” para cada um dos servos.

O código 2 apresenta a função *servoMonitor ()* que imprime as informações referentes a todos os servos. Esta função utiliza a função *printServo ()*, descrita anteriormente, que recebe a angulação e o número do servo correspondente para impressão e termina com a função *line ()* que encerra a linha.

Como alternativa do Monitor Serial, podemos ver em forma de gráficos a variação de angulação dos servomotores através do plotter Serial (ver Figura 4).

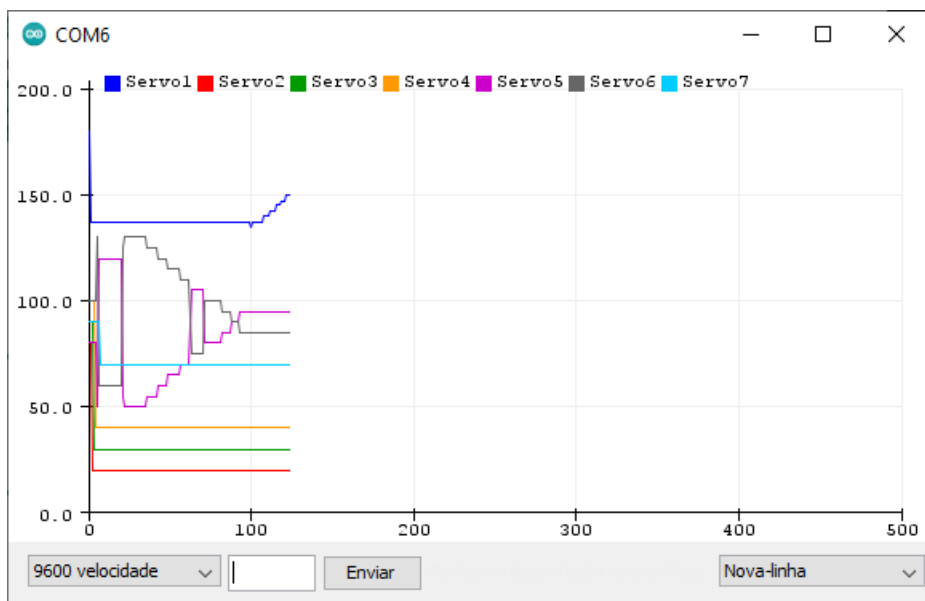


Figura 4: Demonstração do Plotter Serial.

Este gráfico foi elaborado utilizando a função *servoPlotter ()* mostrada no Código 3.

```

void servoPlotter(){
    Serial.print("Servo1:");
    Serial.print(ang1);
    Serial.print(",Servo2:");
    Serial.print(ang2);
    Serial.print(",Servo3:");
    Serial.print(ang3);
    Serial.print(",Servo4:");
    Serial.print(ang4);
    Serial.print(",Servo5:");
    Serial.print(ang5);
    Serial.print(",Servo6:");
    Serial.print(ang6);
    Serial.print(",Servo7:");
    Serial.println(ang7);
}

```

Código 3: Função que imprime cada um dos ângulos dos servos para o Plotter Serial.

Cada variável “ang” representa a angulação de um servomotor. Assim, ao imprimir esses valores no plotter serial, o plotter interpreta cada um e os imprime com cores diferentes conforme ilustrado na legenda.

### 5.3.2 Recebimento de dados via Bluetooth

O recebimento de dados via *Bluetooth* é bem simples. Como exemplificado pelo código a seguir (Código 4):

```

void setup () {
    Serial.begin(9600);
}

void loop () {
    if (Serial.available()) {
        Serial.println(Serial.read());
    }
}

```

Código 4: Recepção Bluetooth simplificada.

Na função *setup ()* ele apenas inicia a comunicação serial na velocidade 9600 bauds, ou seja, 9600 bits por segundo. Na função *loop ()* se o valor estiver disponível na entrada serial, ele irá fazer a leitura e imprimir o valor lido.

### 5.3.3 Recebimento de dados via wi-fi

O recebimento de dados via wi-fi apresenta uma complexidade muito superior à do envio de dados via Bluetooth. Foram realizados testes com o módulo NodeMCU Amica que possui wi-fi integrado. Nesses testes pode-se observar demora superior quando comparado ao Arduino, na configuração inicial do microcontrolador, assim como para fazer o upload dos códigos para o microcontrolador. O código utilizado no teste é uma junção de várias linguagens de programação, o que pode gerar certa confusão ao se misturar comandos em C com as tags do HTML. Mas apesar dessas dificuldades foi possível realizar a transmissão dos dados.

Com o código base obtido de [San18] e [San181] (ver Figura 5) que transmite os dados ao pressionar dois botões, modificou-se o código para que ao invés de transmitir os dados a partir dos botões, fosse transmitido após a mudança de posição de um *slider* (Controle deslizante). Entretanto, a informação só é enviada após soltar o *slider* na nova posição, o que reduz a quantidade de dados que é transmitida.

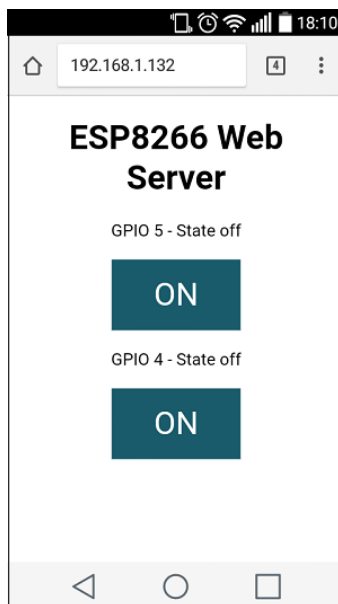


Figura 5: ESP8266 Web Server - Imagem demonstrativa disponível em [San18].

Mesmo com esses pontos negativos, é esperado um estudo mais aprofundado em relação à aplicação wi-fi e que as características ruins sejam amenizadas.

## 5.4 Seleção da linguagem de programação para desenvolvimento de apps

Diante da vasta gama de linguagens de programação disponíveis foi feita uma pesquisa sobre algumas linguagens de programação e os seus benefícios, sendo classificadas quanto



à sua aplicabilidade no atual projeto [Sch19].

Swift – Não apropriado: É uma linguagem específica para aplicativos nativos do Sistema Operacional da Apple, o Apple OS (iOS). O que impede que sejam desenvolvidos apps para Smartphones Android que são o foco do projeto.

Java – Viável e apropriado: É uma linguagem específica para aplicativos nativos do Android, o que permite o uso amplo dos recursos disponíveis nos Smartphones, porém, acaba não sendo prático caso seja visado no futuro o desenvolvimento de um app que aceite também os Smartphones da Apple. Por esse critério foi deixado de lado, porém não se descarta a possibilidade de seu uso em algum projeto futuro.

Kotlin – Viável e apropriado: Possui as mesmas desvantagens do Java.

Ionic e Flutter – Viáveis e apropriados: São opções consideravelmente similares ao React Native em sua funcionalidade. E por isso também poderiam ser utilizadas nesse projeto.

JavaScript – Viável e apropriado: É a linguagem utilizada no *back-end* do app. É a linguagem que em conjunto com o framework React Native, torna o app funcional.

React Native – Apropriado: Permite que um app utilizando da linguagem JavaScript seja convertido para a linguagem nativa tanto do iOS quanto do Android. Também podendo ser considerado que a lógica presente no desenvolvimento se assemelha (e utiliza de conceitos) de linguagens mais consolidadas, sendo elas HTML e CSS (mais utilizadas em desenvolvimento web) e também de React (uma biblioteca de JavaScript para desenvolvimento de interfaces) e JavaScript (linguagem *back-end*). Além disso pode ser desenvolvido utilizando o Expo. Que torna possível programar utilizando React Native e quase que instantaneamente ver o app resultante em um Smartphone conectado à mesma rede wi-fi que tenha o app do Expo. Mesmo com a velocidade o Expo tem suas limitações: ao utilizá-lo não é possível importar bibliotecas (conjunto de código que permitem o acesso a novas funcionalidades) foram necessárias para o projeto. Também tem seu uso consolidado em diversos apps conhecidos como Facebook, Instagram, Discord, dentre outros [Nat21].

Com isso foi decidido o uso do React Native em conjunto com o JavaScript para o desenvolvimento do App do atual projeto.

## 5.5 Aplicativos

Para entender melhor o uso e a sintaxe da linguagem escolhida, foram desenvolvidos diversos apps.

### 5.5.1 Familiarização com React Native

Para poder desenvolver o app, primeiramente foi necessário aprender sobre a linguagem que seria utilizada. Primeiro aprendendo a montar o ambiente de trabalho com os softwares necessários, sendo eles VS Code para programar o app, Android Studio para poder emular um dispositivo Android em que o desenvolvimento do app seria observado e

também o Node.js que disponibiliza algumas ferramentas de desenvolvedor e também o Expo que permite uma rápida atualização através de seu app.

Durante o aprendizado utilizando o Expo, foram desenvolvidos diversos apps com funcionalidades diferentes para se explorar as capacidades da linguagem e entender o seu funcionamento. Foram testados componentes como Bloco de Texto, Imagem, Relógio, Caixa de Texto, e o View (que é um encapsulador de componentes que pode ter seu estilo alterado para poder modelar graficamente o aplicativo).

Após diversos testes com esses componentes e variáveis, percebeu-se uma limitação presente no Expo: quando a biblioteca para o uso do Bluetooth foi adicionada, o Expo não a reconheceu, com isso, foi necessário dispensar o Expo para que o desenvolvimento continuasse.

Utilizando o Node.js para poder atualizar o app visualmente, mesmo que a velocidade de atualização tenha reduzido, não havia mais limitação em relação ao uso da biblioteca necessária. E assim, utilizando apenas o módulo Bluetooth HC-05 conectado ao Arduino pode-se desenvolver o app para que ele enviasse os dados devidamente. Devido à forma como o Arduino recebe os dados, foi necessário primeiramente converter os dados que seriam enviados utilizando a base 64 que é uma forma de compactar os dados enviados em caracteres mais comuns a fim de reduzir a quantidade de bytes necessários para enviar texto através de dispositivos, assim reduzindo o risco de perda de dados por incompatibilidade de caracteres entre diferentes plataformas.

### 5.5.2 *Conexão com dispositivos Bluetooth*

Após desenvolver um app com conexão Bluetooth acompanhando o [Lar20], foi possível desenvolver um aplicativo próprio que conseguisse se conectar ao módulo Bluetooth HC-05 conectado ao Arduino, fato confirmado pela mudança no padrão luminoso emitido pelo módulo. Este aplicativo foi desenvolvido com base em um Workshop [Lar20] (Figura 6).

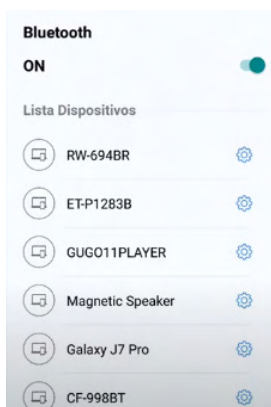


Figura 6: Aplicativo desenvolvido no WorkShop - Imagem demonstrativa disponível em [Lar201].

### 5.5.3 Envio de dados via Bluetooth com botões e sliders

Após pesquisas na documentação de bibliotecas de comunicação Bluetooth [Dav20], [Dav21], pôde-se entender e utilizar as funções necessárias para o envio de dados via Bluetooth.

Baseado no que foi aprendido com o app de treinamento, e na pesquisa das documentações, foi possível alterar algumas funções já existentes para poder enviar dos dados utilizando a biblioteca *react-native-base64* [era20] para converter os dados enviados para a base 64 já citada anteriormente.

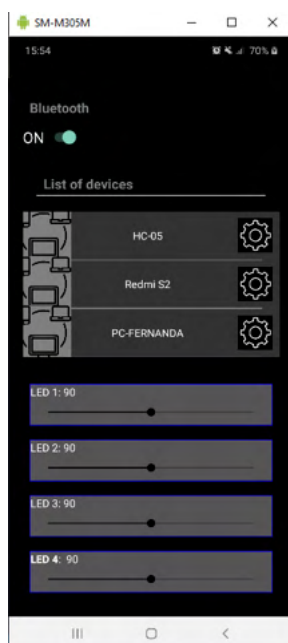


Figura 7: Um dos primeiros layouts do app de envio de dados por slider.

O app mostrado na Figura 7 permite ligar e desligar o Bluetooth no dispositivo, se conectar a um dispositivo disponível e enviar para ele os valores dos *sliders* quando eles eram movimentados. Com isso foi possível se conectar ao módulo HC-05 e enviar os dados apropriadamente.

Vale ressaltar que o aplicativo é capaz de cumprir com o objetivo do projeto da pesquisa, conseguindo se conectar através de uma tecnologia de comunicação remota à um dispositivo microcontrolado capaz de movimentar os servomotores do novo braço robótico.

## 5.6 Braço robótico montado

A montagem do braço robótico foi bastante trabalhosa e durou em torno de 10 horas. De um modo geral não foi muito complicado, porém a falta de instruções claras

presentes no manual reduziu o rendimento. Dessa forma, foi inevitável que acontecessem erros de montagem. Outros problemas foram a existência de peças que não encaixavam adequadamente e necessitaram de um pouco de força para se fixarem devidamente, e também alguns parafusos que eram mais longos do que necessário e outros que eram mais curtos do que o necessário.

Optou-se por um modelo de braço robótico com 5 graus de liberdade que tem 5 servomotores “*Servo Motor MG996R Tower Pro*” [ETC14] que são mais potentes e trazem maior resistência ao peso da estrutura do que os servomotores “*Micro Servo 9g SG90 TowerPro*” [Pro14], [Flo21], presentes no primeiro protótipo construído em pesquisa anterior.

A Figura 8 mostra o segundo protótipo devidamente montado.

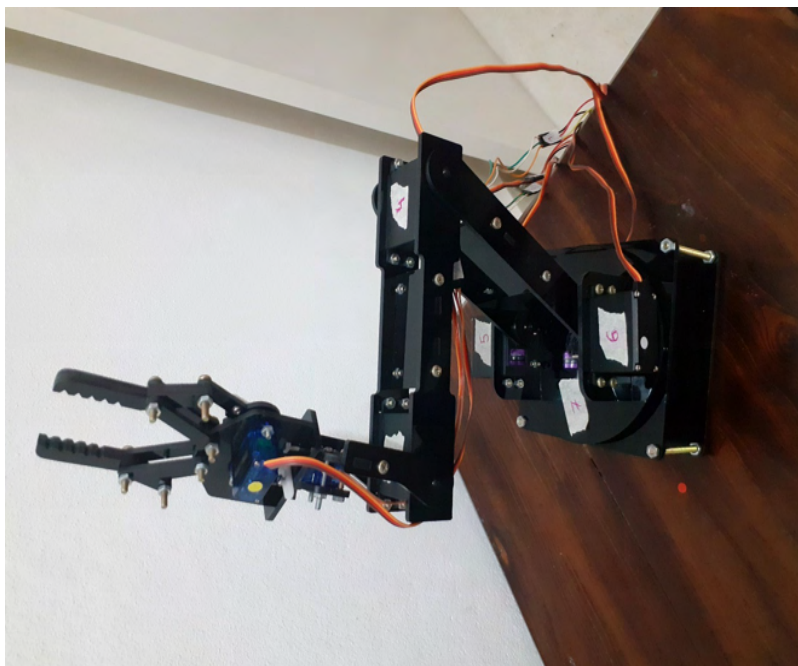


Figura 8: Braço robótico montado.

## 5.7 Movimentação do braço robótico com controle via Bluetooth

Após o desenvolvimento do app que enviava os valores dos *sliders* apropriadamente para o HC-05 e da conclusão da montagem do braço robótico, os diferentes segmentos do projeto foram conectados, resultando na movimentação esperada do braço robótico, necessitando apenas de alguns ajustes em relação à angulação máxima de cada servo motor. Após esses ajustes, o braço robótico deveria estar completamente funcional.

Após algumas tentativas percebeu-se que após determinada angulação, os servos não eram capazes de suportar o peso da estrutura e acabavam deixando a estrutura descer

lentamente. Essa situação inesperada não se encontra resolvida e espera-se que uma solução seja encontrada na continuação da pesquisa.

## 5.8 Testes do Bluetooth com outros microcontroladores

Como citado anteriormente, com a conclusão do objetivo do projeto utilizando o Arduino UNO com o módulo Bluetooth HC-05 e o Sensor Shield v5.0 [Ele20]. Passou-se a testar os outros microcontroladores utilizando o mesmo app. Após alguns ajustes nas configurações do Arduino IDE o HC-05 pode ser conectado ao NodeMCU Amica sem maiores problemas e os valores recebidos puderam ser lidos adequadamente.

Após alguns testes bem sucedidos os valores recebidos pelo HC-05 mudaram de padrão aparentemente sem motivo. Passaram a serem recebidos os seguintes valores:

192      5      0      0      250      192

Após variar os valores enviados e a quantidades de caracteres, percebeu-se que os valores 192 e 0 não se alteravam. Apenas o segundo e penúltimo valor se alteraram. Sendo o segundo referente a quantidade de caracteres e o penúltimo sendo o resultado do cálculo “255-quantidade de caracteres”.

Após novos testes com quantidades ainda maiores de caracteres, percebeu-se que quando a mensagem enviada era maior do que 255 caracteres, o primeiro 0 recebido se tornava 1. Se fosse maior que  $2 \times 255$  (510) caracteres o primeiro 0 recebido se tornaria 2. Sendo assim, estima-se que caso a mensagem seja maior que  $255 \times 255$ , o segundo 0 se torne, permitindo então uma mensagem de tamanho total  $255 \times 255 \times 255$  (16.581.375) caracteres. Alguns valores também apresentaram comportamentos inesperados, e estão passíveis de maiores análises. De uma forma geral a quantidade de caracteres é enviada, logo o dado aparentemente ainda está sendo enviado, porém não está sendo lido ou interpretado corretamente.

Quanto a solução do bug, foram testados diversos códigos diferentes, em diversos microcontroladores, com diversos apps diferentes e diversos softwares de terminal. Em todos eles apresentaram o mesmo bug que antes não ocorria. Com isso cogita-se a possibilidade de falha no hardware do HC-05. Porém essa afirmação não é assertiva, visto que não houve nenhuma modificação externa aparente que possa ter causado essa mudança, mesmo assim, espera-se que posteriormente um outro módulo Bluetooth seja testado para comprovar a hipótese.

## CONCLUSÕES

As etapas do plano de trabalho foram em sua maioria concluídas com êxito, restando apenas resolver o bug do módulo *Bluetooth* HC-05, a implementação da conexão via wi-fi, analisar e montar o circuito elétrico necessário para manter ativos os servomotores do braço robótico e aplicar as melhorias propostas ao app.

Este relatório apresenta informações sobre diferentes tecnologias de comunicação remota, restringindo o filtro para as que fossem aplicáveis à microcontroladores e à utilização pelo Smartphone. Sendo então escolhidas as tecnologias wi-fi e Bluetooth para que fossem testadas.

Houve também pesquisa sobre diferentes linguagens de programação que permitissem o desenvolvimento de aplicativos. Dentre várias opções, foi decidido utilizar a linguagem React Native devido a praticidade proporcionada pelo Expo.

Foram pesquisados diversos modelos de braços robóticos, com isso foi escolhido um modelo de acrílico com 5 graus de liberdade que foi adquirido e montado. As plataformas microcontroladas escolhida foi o Arduino por causa da disponibilidade do módulo *Sensor Shield* que permitia a gestão de vários servomotores. Mesmo que também tenha sido testado o uso do módulo NodeMCU para proporcionar conexão via wi-fi.

Foi desenvolvido um aplicativo para controle do braço robótico via Bluetooth que permitiu controlar o braço robótico com precisão, e também um site que permitiu enviar informações via wi-fi para o NodeMCU.

Diversos testes foram realizados para avaliar o desempenho das conexões disponíveis, e também o funcionamento do braço robótico. Com isso foi constatado que a conexão via Bluetooth se mostrou suficientemente rápida e precisa, porém apresentou um bug durante os testes que impossibilitou o uso do módulo (Um novo módulo foi encomendado para descartar possível falha física). A conexão wi-fi aparenta estar limitada ao envio de dados apenas quando o slider for solto, porém mais testes podem ser feitos para confirmar essa análise. O braço robótico não conseguiu suportar o próprio peso. Suspeita-se que a falta de força se deva à fonte de energia pois o uso de dois servomotores MG996R demonstram ser mais do que suficiente para manter firme o braço como um todo. Também foi visto que o braço robótico conseguiu ser muito responsivo em relação aos dados enviados pelo Bluetooth. Porém ele tem o potencial de tombar devido aos movimentos bruscos, precisando então ser preso à uma base temporária enquanto as melhorias descritas logo abaixo não são implementadas.

## 6 | REFERÊNCIAS

[aKa18] a, K. R., b, T. A., & Abbas, N. (27 de março de 2018). **Kinematic analysis and geometrical improvement of an industrial**. *Science Direct*, 30, pp. 218-223. doi: <https://doi.org/10.1016/j.jksues.2018.03.005> Citado na página 1.

[Ard21] Arduino. (2021). *environment*. (SM, Editor) Fonte: Arduino: <https://www.arduino.cc/en/guide/environment> Citado na página 2.

[Ban19] Bane, A., Daoui, M., Bouzeffrane, S., & Muhlethaler, P. (Abril de 2019). **NDN-over-ZigBee: A ZigBee support for Named Data Networking**. *Science Direct*, 93, pp. 792-798. doi: <https://doi.org/10.1016/j.future.2017.09.053> Citado na página 5.

[Biø19] Bjørn-Hansen, A., Grønli, T.-M., Ghinea, G., & Alouneh, S. (03 de janeiro de 2019).

**An Empirical Study of Cross-Platform Mobile.** (G. Canfora, Ed.) *Hindawi*, 2019, p. 12. doi: <https://doi.org/10.1155/2019/5743892> Citado na página 2.

[Cap21] Capes. (2021). Buscar Assunto. Fonte: Periódicos Capes: <https://www-periodicos-capes-gov-br.ez81.periodicos.capes.gov.br/index.php> Citado na página 2.

[Dav20] Davidson, K. (18 de novembro de 2020). *react-native-bluetooth-classic*. Fonte: GitHub: <https://github.com/kenjdavidson/react-native-bluetooth-classic> Citado na página 12.

[Dav21] Davidson, K. (s.d.). **React Native Bluetooth Classic**. Acesso em 2021, disponível em [kenjdavidson.com/react-native-bluetooth-classic/](https://kenjdavidson.com/react-native-bluetooth-classic/) Citado na página 12.

[Dun17] Dunbar, B. (03 de março de 2017). *Curiosity Rover*. (T. Greicius, Editor) Acesso em 2021, disponível em NASA: [https://www.nasa.gov/mission\\_pages/msl/overview/index.html](https://www.nasa.gov/mission_pages/msl/overview/index.html) Citado na página 2.

[Dun171] Dunbar, B. (03 de março de 2017). *Spirit and Opportunity*. (T. Greicius, Editor) Acesso em 2021, disponível em NASA: [https://www.nasa.gov/mission\\_pages/mer/overview/index.html](https://www.nasa.gov/mission_pages/mer/overview/index.html) Citado na página 2.

[Ele20] Electronics, E. G. (2021). Fonte: Curto Circuito: [https://curtocircuito.com.br/datasheet/arduino\\_sensor\\_shield.pdf](https://curtocircuito.com.br/datasheet/arduino_sensor_shield.pdf) Citado na página 14.

[era20] eranbo. (dezembro de 2020). *react-native-base64*. Acesso em 2021, disponível em npmjs: <https://www.npmjs.com/package/react-native-base64> Citado na página 12.

[ETC14] ETC. (2014). Acesso em 2021, disponível em [datasheetspdf.com/pdf/942981/ETC/MG996R/1](https://datasheetspdf.com/pdf/942981/ETC/MG996R/1) Citado na página 13.

[FAQ21] FAQ. (31 de março de 2021). Fonte: Visual Studio Code: <https://code.visualstudio.com/docs/supporting/faq> Citado na página 3.

[Flo21] Flop, F. (s.d.). *micro-servo-9g-sg90-towerpro*. Acesso em 2021, disponível em Filipe Flop: <https://www.filipeflop.com/produto/micro-servo-9g-sg90-towerpro/> Citado na página 13.

[Jaw18] Jaworski, T. S., & Noichl, S. (03 de abril de 2018). **Evaluation of Cross-Platform App Development using React Native**. doi: <http://doi.org/10.18154/RWTH-2018-223392> Citado na página 2.

[Koy18] Koyanagi, F. (09 de janeiro de 2018). **ESP32 Longa Distância - LoRaWan**. Fonte: Fernando K: <https://www.fernandok.com/2018/01/esp32-longa-distancia-lorawan.html> Citado na página 5.

[Koy181] Koyanagi, F. (31 de julho de 2018). **Protocolo CAN - Yes, We Can!** Fonte: Fernando K: <https://www.fernandok.com/2018/07/protocolo-can-yes-we-can.html> Citado na página 5.

[Koy21] Koyanagi, F. (2021). **Tutoriais, Tecnologias, Tendências**. Fonte: Fernando K Tecnologias: <https://www.fernandok.com/> Citado na página 2.

[Lar201] Lara, C. (1 de março de 2020). **7 - React Native em espanhol I Utilizando la dependencia bluetooth**. Acesso em 2021, disponível em Youtube: <https://youtu.be/LlIkwiK4hz8?list=PLTYm84ujubwL3aYYWkU9FoZBrkW9GFy2G&t=469> Citado na página 12.

[Lar20] Lara, C. (29 de fevereiro de 2020). **WorkShop React Native + Bluetooth**. Acesso em 2021, disponível em YouTube: <https://youtube.com/playlist?list=PLTYm84ujubwL3aYYWkU9FoZBrkW9GFy2G> Citado na página 12.

[Lee18] Leenheer, N. (10 de janeiro de 2018). **Fun With Bluetooth**. Munique, Baviera, Alemanha. Acesso em 2021, disponível em Youtube: <https://youtu.be/XDc5HUVMI5U> Citado na página 2.

[Lee20] Lee, H.-W. (19 de junho de 2020). **The Study of Mechanical Arm and Intelligent Robot**. *IEEE access*, 8, pp. 119624-119634. doi: <https://doi.org/10.1109/ACCESS.2020.3003807> Citado na página 1.

[Nat21] Native, R. (2021). Acesso em 2021, disponível em React Native: <https://reactnative.dev/> Citado na página 10.

[Nik19] Nikravan, M., Movaghar, A., & Hosseinzadeh, M. (11 de maio de 2019). Springer. **Peer-to-Peer Networking and Applications**, 12, pp. 209-226. doi: <https://doi.org/10.1007/s12083-018-0659-8> Citado na página 5.

[Plu18] Plus, D. (2018). *Curso de React Native*. Acesso em 2021, disponível em Youtube: <https://youtube.com/playlist?list=PLxF2lyHGcERApnjQPgeeElzJdGurraMW> Citado na página 2.

[Plu19] Plus, D. (2019). *Curso de React Native*. Acesso em 2021, disponível em Developer Plus: <https://developerplus.com.br/category/curso-de-react-native/> Citado na página 2.

[Pro14] Pro, T. (2014). Acesso em 2021, disponível em datasheetpdf: <https://datasheetpdf.com/pdf/791970/TowerPro/SG90/1> Citado na página 13.

[San181] Santos, R. (20 de fevereiro de 2018). **Build an ESP8266 Web Server with Arduino IDE - Code and Schematics**. Acesso em 2021, disponível em YouTube: [https://www.youtube.com/watch?v=dWM4p\\_KaTHY](https://www.youtube.com/watch?v=dWM4p_KaTHY) Citado na página 9.

[San14] Santos, R. L. (2014). **Internet das Coisas e 6LoWPAN**. Monografia, Universidade Tecnológica Federal do Paraná, Curitiba. Acesso em 02 de Abril de 2021, disponível em [http://repositorio.utfpr.edu.br/jspui/bitstream/1/17312/2/CT\\_GESER\\_V\\_2014\\_12.pdf](http://repositorio.utfpr.edu.br/jspui/bitstream/1/17312/2/CT_GESER_V_2014_12.pdf) Citado na página 1.

[San18] Santos, R., & Santos, S. (20 de fevereiro de 2018). **ESP8266 Web Server with Arduino IDE**. Acesso em 2021, disponível em Random Nerd Tutorials: <https://randomnerdtutorials.com/esp8266-web-server-with-arduino-ide/> Citado na página 9.

[Sch19] Schwarzmüller, M. (14 de agosto de 2019). **Which one is best for you? Flutter, React Native, Ionic or NativeScript?** Acesso em 2021, disponível em YouTube: <https://youtu.be/PKRXbLnfXXk> Citado na página 10.

[Sea21] Seat, R. (2021). *React Native*. Fonte: Next Level Week: <https://nextlevelweek.com/>



[Ste18] Stevan Jr., S. (2018). **IoT Internet das coisas Fundamentos e aplicações em Arduino e NodeMCU**. (S. C. Ferreira, Ed.) São Paulo, São Paulo, Brasil: Saraiva. Acesso em 2021  
Citado na página 1.

[Tre21] *Treinamentos*. (2021). Fonte: Periódicos Capes: [https://www-periodicos-capes-gov-br.ez81.periodicos.capes.gov.br/index.php?option=com\\_ptreinamentos&Itemid=306](https://www-periodicos-capes-gov-br.ez81.periodicos.capes.gov.br/index.php?option=com_ptreinamentos&Itemid=306) Citado na página 2.

[Vis21] (2021). Fonte: Visual Studio Code: <https://code.visualstudio.com/> Citado na página 3.

[Wil21] William. (2021). *DroneBot Workshop*. Fonte: DroneBot Workshop: <https://dronebotworkshop.com/> Citado na página 2.

## 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](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)

COLEÇÃO

# DESAFIOS DAS ENGENHARIAS:

ENGENHARIA DE COMPUTAÇÃO 3

- 🌐 [www.atenaeditora.com.br](http://www.atenaeditora.com.br)
- ✉ [contato@atenaeditora.com.br](mailto:contato@atenaeditora.com.br)
- 📷 @atenaeditora
- 📘 [www.facebook.com/atenaeditora.com.br](https://www.facebook.com/atenaeditora.com.br)