



Universidade Federal do Espírito Santo  
Centro de Ciências Exatas  
Programa de Pós-Graduação em Matemática em Rede Nacional

**DISSERTAÇÃO DE MESTRADO**

**APROPRIAÇÃO DO PENSAMENTO COMPUTACIONAL E DA  
ROBÓTICA EDUCACIONAL PARA UM CURRÍCULO  
ALINHADO ÀS NOVAS TENDÊNCIAS EM TECNOLOGIAS  
EDUCACIONAIS**

**MÁRCIO PETERS**



**PROFMAT**

Vitória - ES  
2023



**MÁRCIO PETERS**

**APROPRIAÇÃO DO PENSAMENTO COMPUTACIONAL E DA  
ROBÓTICA EDUCACIONAL PARA UM CURRÍCULO  
ALINHADO ÀS NOVAS TENDÊNCIAS EM TECNOLOGIAS  
EDUCACIONAIS**

Dissertação de mestrado  
apresentada ao PROFMAT como  
parte dos requisitos exigidos para a  
obtenção do título de Mestre em  
Matemática

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO  
MESTRADO PROFISSIONAL EM MATEMÁTICA EM REDE NACIONAL



**PROFMAT**

Orientador: Prof. Dr. Fábio Corrêa de Castro

Vitória - ES  
2023

**MÁRCIO PETERS**

**APROPRIAÇÃO DO PENSAMENTO COMPUTACIONAL E DA ROBÓTICA  
EDUCACIONAL PARA UM CURRÍCULO ALINHADO ÀS NOVAS  
TENDÊNCIAS EM TECNOLOGIAS EDUCACIONAIS**

Dissertação de mestrado apresentada ao  
PROFMAT como parte dos requisitos  
exigidos para a obtenção do título de  
Mestre em Matemática

Trabalho aprovado, Vitória, ES 15 de junho de 2023.

**Prof. Dr. Fábio Corrêa de Castro**  
Universidade Federal do Espírito Santo  
Orientador

**Prof. Dr. Alcebiades Dal Col Júnior**  
Universidade Federal do Espírito Santo  
Membro Interno

**Prof. Dr. Carlos Henrique Alexandrino**  
Membro Externo

Vitória - ES  
2023

Dedico primeiramente a Deus, que em sua infinita graça nos deu o dom da vida, permitindo que este trabalho acontecesse. Dedico à minha mãe Edimira, que me concebeu, a meus filhos, Luiz Henrique e Lusiane, que são as razões da minha vida e, também, à minha namorada e esposa Eliana (Lanna) e a todos que estiveram comigo nesse processo, com paciência e compreensão nos momentos de trabalho e pesquisas.

## **AGRADECIMENTOS**

Agradeço a Deus que me auxilia em todos os momentos de minha vida e que nunca nos abandona. Também agradeço a meus colegas de sala do mestrado, em especial aos amigos Alejandro, Daniel, Leandro, Vinicius e demais pelos momentos juntos de troca de conhecimento nos grupos de estudos e aos professores do programa que nos ensinaram com muito amor e carinho que demonstram pela linda profissão de professores.

Agradeço a meu orientador, Professor Fábio, que me auxiliou nas descobertas e também pelo apoio nas pesquisas.

## RESUMO

Com base na BNCC (Base Nacional Comum Curricular), o objetivo deste trabalho é fomentar a robótica educacional e o pensamento computacional no ensino de matemática. Considerando a ausência do tema no Currículo do ES, para fomentar a prática do tema, com base em especialistas, propomos atividades relevantes que vão auxiliar os professores em seu trabalho, dentro e fora de sala de aula. Para este fim, criamos uma série de seis circuitos de aprendizagem compostos por oficinas didáticas. Em sequência, os circuitos foram pensados de tal maneira que abordam que proporciona o desenvolvimento de habilidades do pensamento computacional e da robótica educacional no ensino da matemática com o uso das ferramentas “X Logo”, Code.org, Tinkercad, Arduino, “App Inventor” e atividades de computação desplugadas, para o caso do não acesso a computador ou similares. Isto culminou em uma formação de profissionais da educação em parceria com a Secretaria de Educação do Estado do Espírito Santo.

**Palavras-chave:** Matemática. BNCC. Currículo. Pensamento Computacional (PC). Robótica. Robótica educacional. Arduino. Computação plugada e desplugada.

## ABSTRACT

Based on the BNCC (Base Nacional Comum Curricular), the objective of this work is to promote educational robotics and computational thinking in mathematics teaching. Considering the absence of the theme in the ES Curriculum, to encourage the practice of the theme, based on specialists, we propose relevant activities that will help teachers in their work, inside and outside the classroom. To this end, we created a series of six learning circuits composed of didactic workshops. In sequence, the circuits were designed in such a way that they approach that provide the development of computational thinking skills and educational robotics in the teaching of mathematics with the use of the tools “X Logo”, Code.org, Tinkercad, Arduino, “App Inventor” and unplugged computing activities, in case of not having access to a computer or similar. This culminated in training for education professionals in partnership with the Secretary of Education of the State of Espírito Santo.

**Keywords:** Mathematics. BNCC. Curriculum. Computational Thinking (PC). Robotics. Educational robotics. Arduino. Plugged and unplugged computing.



## Lista de ilustrações

Número	Título	Página
Figura 1	Evolução da computação	27
Figura 2	Metodologias ativas de aprendizagem no século XXI	28
Figura 3	Características do pensamento computacional	31
Figura 4	Resolução de problemas e suas etapas	33
Figura 5	Fluxo da computação na educação	34
Figura 6	Computação desplugada	40
Figura 7	Estudio Code.org	42
Figura 8	Tinkercad	43
Figura 9	BNCC e as competências gerais	50
Figura 10	Habilidades educacionais para o século XXI	52
Figura 11	Letramento digital	57
Figura 12	Eixo das diretrizes da SBC 2018	58
Figura 13	Códigos identificados na BNCC, conforme a área de conhecimento.	59
Figura 14	Conceitos fundamentais das tecnologias digitais	60
Figura 15	Principais símbolos dos fluxogramas	63
Figura 16	Entrada de dados para fluxogramas	63
Figura 17	Fluxograma para resolução do cubo mágico	64
Figura 18	Fluxograma para resolução do cubo mágico, tradução	65
Figura 19	Fluxograma para resolução de equação do 2º grau	65
Figura 20	Fluxograma construção de um quadrado	66
Figura 21	Fluxograma para imprimir números pares positivos menores que N	66
Figura 22	Fluxograma de uma IDE	68
Figura 23	Índice da comunidade de programação TIOBE	70
Figura 24	Code.org desafio Angry Birds	72
Figura 25	Mascote Scratch	73
Figura 26	PictoBlox	74
Figura 27	Fractais criados com a linguagem Logo	75
Figura 28	Robô Curiosity	79
Figura 29	Scratch, mascote e blocos	87
Figura 30	Scratch e Arduino	88

<b>Figura 31</b>	<b>Tinkercad</b>	<b>89</b>
<b>Figura 32</b>	<b>Tinkercad e a placa Arduino</b>	<b>90</b>
<b>Figura 33</b>	<b>Mapa mental dos componentes básicos da robótica</b>	<b>92</b>
<b>Figura 34</b>	<b>LED e seus principais elementos</b>	<b>93</b>
<b>Figura 35</b>	<b>LED RGB</b>	<b>94</b>
<b>Figura 36</b>	<b>Placa Protoboard</b>	<b>94</b>
<b>Figura 37</b>	<b>Resistores</b>	<b>95</b>
<b>Figura 38</b>	<b>Potenciômetros</b>	<b>96</b>
<b>Figura 39</b>	<b>Motores de corrente contínua</b>	<b>97</b>
<b>Figura 40</b>	<b>Motor DC</b>	<b>98</b>
<b>Figura 41</b>	<b>Jumpers</b>	<b>98</b>
<b>Figura 42</b>	<b>MIT App Inventor</b>	<b>99</b>
<b>Figura 43</b>	<b>Área de trabalho do App Inventor</b>	<b>100</b>
<b>Figura 44</b>	<b>Placa Arduino UNO</b>	<b>101</b>
<b>Figura 45</b>	<b>Ambiente editor do Arduino</b>	<b>102</b>
<b>Figura 46</b>	<b>Arduino Pro Mini</b>	<b>102</b>
<b>Figura 47</b>	<b>Arduino Fio</b>	<b>103</b>
<b>Figura 48</b>	<b>Arduino Micro</b>	<b>103</b>
<b>Figura 49</b>	<b>Arduino Nano</b>	<b>104</b>
<b>Figura 50</b>	<b>Identificação Arduino Nano</b>	<b>105</b>
<b>Figura 51</b>	<b>Identificação Arduino Uno</b>	<b>105</b>
<b>Figura 52</b>	<b>Ambiente X LOGO</b>	<b>108</b>
<b>Figura 53</b>	<b>Desenhando um quadrado com X LOGO</b>	<b>110</b>
<b>Figura 54</b>	<b>Desenhando um triângulo com X LOGO</b>	<b>111</b>
<b>Figura 55</b>	<b>Desenhando um retângulo com X LOGO</b>	<b>112</b>
<b>Figura 56</b>	<b>Desenhando com X LOGO usando comando repita</b>	<b>113</b>
<b>Figura 57</b>	<b>Comando aprenda e repita</b>	<b>115</b>
<b>Figura 58</b>	<b>Comando quadrado 100</b>	<b>116</b>
<b>Figura 59</b>	<b>Comando contevezes</b>	<b>117</b>
<b>Figura 60</b>	<b>Lições no code.org</b>	<b>118</b>
<b>Figura 61</b>	<b>Códigos do code.org</b>	<b>119</b>
<b>Figura 62</b>	<b>Entradas e saídas do Arduino Uno</b>	<b>120</b>
<b>Figura 63</b>	<b>Área de trabalho do Tinkercad</b>	<b>121</b>
<b>Figura 64</b>	<b>Circuitos no Tinkercad</b>	<b>122</b>
<b>Figura 65</b>	<b>Simulando o projeto “Piscar”</b>	<b>122</b>
<b>Figura 66</b>	<b>Código em blocos do projeto “Piscar”</b>	<b>123</b>

<b>Figura 67</b>	<b>Código em texto do projeto “Piscar”</b>	<b>124</b>
<b>Figura 68</b>	<b>Iniciar simulação do projeto “Piscar”</b>	<b>124</b>
<b>Figura 69</b>	<b>Placa Arduino conectada ao computador</b>	<b>126</b>
<b>Figura 70</b>	<b>Placas protoboard</b>	<b>127</b>
<b>Figura 71</b>	<b>Kit de componentes eletrônicos para Arduino</b>	<b>129</b>
<b>Figura 72</b>	<b>Circuitos elétricos ligados na placa protoboard</b>	<b>130</b>
<b>Figura 73</b>	<b>Circuito elétrico ligado no pino GND e no 5V do Arduino</b>	<b>131</b>
<b>Figura 74</b>	<b>Circuito elétrico alimentado direto via USB e outro na fonte</b>	<b>131</b>
<b>Figura 75</b>	<b>Simulação do circuito Blink LED no Tinkercad</b>	<b>132</b>
<b>Figura 76</b>	<b>LED com tempo programado na prática</b>	<b>134</b>
<b>Figura 77</b>	<b>Circuito elétrico simples com chave interruptor</b>	<b>135</b>
<b>Figura 78</b>	<b>LED controlado com o acionamento de um botão</b>	<b>136</b>
<b>Figura 79</b>	<b>Sketch do desafio sinalizador de garagem</b>	<b>138</b>
<b>Figura 80</b>	<b>Sketch do desafio para acender 2 LEDs</b>	<b>139</b>
<b>Figura 81</b>	<b>Cinco LEDs piscando de forma alternada</b>	<b>142</b>
<b>Figura 82</b>	<b>Potenciômetro e o Arduino</b>	<b>143</b>
<b>Figura 83</b>	<b>Potenciômetro controlando LEDs</b>	<b>144</b>
<b>Figura 84</b>	<b>Motor com potenciômetro</b>	<b>146</b>
<b>Figura 85</b>	<b>Sketch para o projeto sensor de luminosidade</b>	<b>148</b>
<b>Figura 86</b>	<b>Sketch para o projeto sensor de luminosidade mais buzzer</b>	<b>149</b>
<b>Figura 87</b>	<b>Sketch para o projeto sensor infravermelho</b>	<b>150</b>
<b>Figura 88</b>	<b>Sketch para o projeto alarme de segurança</b>	<b>151</b>
<b>Figura 89</b>	<b>Sketch para o projeto carro seguidor de linha 2WD</b>	<b>153</b>
<b>Figura 90</b>	<b>Sketch para o projeto carro contorno de obstáculos 2WD</b>	<b>155</b>
<b>Figura 91</b>	<b>Sketch para o projeto carro 2WD com LCD</b>	<b>157</b>
<b>Figura 92</b>	<b>Ambiente MIT APP Inventor</b>	<b>161</b>
<b>Figura 93</b>	<b>Área de trabalho do APP Inventor</b>	<b>162</b>
<b>Figura 94</b>	<b>Blocos de códigos do APP Inventor</b>	<b>163</b>
<b>Figura 95</b>	<b>Qr Code do MIT AI2</b>	<b>163</b>
<b>Figura 96</b>	<b>App comandos via Bluetooth</b>	<b>165</b>
<b>Figura 97</b>	<b>App controlando LED via Bluetooth</b>	<b>167</b>
<b>Figura 98</b>	<b>App controlando LED via Bluetooth, layout</b>	<b>168</b>
<b>Figura 99</b>	<b>App controlando LED via Bluetooth, botões</b>	<b>169</b>
<b>Figura 100</b>	<b>App controlando LED via Bluetooth, propriedades</b>	<b>169</b>
<b>Figura 101</b>	<b>App controlando LED via Bluetooth, finalizado</b>	<b>170</b>
<b>Figura 102</b>	<b>Blocos lógicos do App Inventor</b>	<b>171</b>

<b>Figura 103</b>	<b>Blocos lógicos do App Inventor, notas 1</b>	<b>171</b>
<b>Figura 104</b>	<b>Blocos lógicos do App Inventor, screen 1</b>	<b>172</b>
<b>Figura 105</b>	<b>Blocos lógicos do App Inventor, bluetooth</b>	<b>172</b>
<b>Figura 106</b>	<b>Blocos lógicos do App Inventor, itens suspensos</b>	<b>173</b>
<b>Figura 107</b>	<b>Blocos lógicos do App Inventor, busca do dispositivo</b>	<b>174</b>
<b>Figura 108</b>	<b>Blocos lógicos do App Inventor, escolhe lista</b>	<b>174</b>
<b>Figura 109</b>	<b>Blocos lógicos do App Inventor, botões</b>	<b>175</b>
<b>Figura 110</b>	<b>Blocos lógicos do App Inventor, checkbox</b>	<b>176</b>
<b>Figura 111</b>	<b>Blocos lógicos do App Inventor, duplicar botões</b>	<b>176</b>
<b>Figura 112</b>	<b>Sketch do projeto LED controlado por Bluetooth</b>	<b>177</b>
<b>Figura 113</b>	<b>Sketch do projeto Motor DC controlado por Bluetooth</b>	<b>179</b>
<b>Figura 114</b>	<b>Cartões controle remoto, recorte 1</b>	<b>183</b>
<b>Figura 115</b>	<b>Cartões controle remoto, recorte 2</b>	<b>184</b>
<b>Figura 116</b>	<b>Cartão colorindo com números</b>	<b>185</b>
<b>Figura 117</b>	<b>Cartão crie uma imagem</b>	<b>186</b>
<b>Figura 118</b>	<b>Blocos Scratch - Passeio da tartaruga</b>	<b>187</b>
<b>Figura 119</b>	<b>Passeio da tartaruga</b>	<b>188</b>
<b>Figura 120</b>	<b>Formação Ensino Maker, Apresentação</b>	<b>189</b>
<b>Figura 121</b>	<b>Formação Ensino Maker, tópico 1</b>	<b>190</b>
<b>Figura 122</b>	<b>Formação Ensino Maker, tópico 2</b>	<b>190</b>
<b>Figura 123</b>	<b>Formação Ensino Maker, tópico 3</b>	<b>191</b>
<b>Figura 124</b>	<b>Formação Ensino Maker, tópico 4</b>	<b>191</b>
<b>Figura 125</b>	<b>Formação Ensino Maker, avaliação da formação</b>	<b>192</b>

## Sumário

<b>Lista de ilustrações</b>	<b>09</b>
<b>Sumário</b>	<b>13</b>
<b>1. Introdução</b>	<b>17</b>
<b>2. Pensamento Computacional, apropriação do conceito à prática</b>	<b>24</b>
2.1 Pensamento Computacional (PC)	24
2.1.1 Reflexões de alguns autores sobre o Pensamento Computacional	25
2.1.2 Pensamento Computacional (PC) e sua linha do tempo	26
2.1.3 Metodologias utilizadas para trabalhar o PC	27
2.1.4 Preciso aprender e ensinar os aprendizes a escreverem códigos para desenvolver as habilidades do PC?	30
2.1.5 Pensamento computacional na resolução de problemas	32
2.2 Computação desplugada	39
2.3 Exemplos de como a computação pode ser mais fácil de ensinar e aprender do que imaginamos	41
2.4 Habilidades do profissional do futuro	43
<b>3. A (não) abordagem das tecnologias digitais na BNCC e no Currículo do Espírito Santo e o letramento digital</b>	<b>46</b>
3.1 A BNCC e os necessários apontamentos para o desenvolvimento das habilidades do PC	46
3.2 O Currículo do ES e os necessários apontamentos para o desenvolvimento das habilidades do PC	52
3.3 Letramento e/ou educação digital	55
<b>4. Fluxogramas, algoritmos e linguagens de programação</b>	<b>61</b>
4.1 Fluxogramas	61
4.2 Utilização e aplicações de fluxogramas	64
4.3 Algoritmos	67
4.4 Linguagens de programação e a sua evolução	68
4.5 Tipos de linguagens de programação	71

4.5.1 Linguagem em bloco	72
4.5.2 Code.org (Angry Birds)	72
4.5.3 Scratch	73
4.5.4 Pictoblox	74
4.5.5 X-Logo	75
4.5.6 HTML	76
4.5.7 Pascal	76
4.5.8 Linguagem C	77
<b>5. Robótica Educacional, Arduino e componentes eletrônicos</b>	<b>78</b>
5.1 A Robótica	78
5.2 Robótica educacional	82
5.3 Scratch	86
5.4 Tinkercad	88
5.5 Eletrônica básica	90
5.5.1 Componentes eletrônicos básicos para a robótica	91
5.5.2 Placas de circuitos eletrônicos	92
5.5.3 LED ou lâmpadas	93
5.5.4 Placa protoboard	94
5.5.5 Resistores	95
5.5.6 Potenciômetro	95
5.5.7 Motor de passo x motor DC	96
5.5.8 Jumpers	98
5.6 MIT APP Inventor	99
5.7 Arduino	100
5.7.1 Modelos e tipos de arduino	102
<b>6. Proposta didática - Circuitos de aprendizagem com oficinas didáticas plugadas e desplugadas</b>	<b>106</b>
6.1 Computação Plugada	107
6.1.1 Circuito de aprendizagem X Logo	107

6.1.1.1 Oficina didática - Instalação do X Logo	107
6.1.1.2 Oficina didática - Desenhar um quadrado	110
6.1.1.3 Oficina didática - Desenhar um triângulo	111
6.1.1.4 Oficina didática - Desenhar um retângulo	112
6.1.1.5 Oficina didática - Desenhar o Círculo usando procedimentos	114
6.1.2 Circuito de aprendizagem - Code.org	118
6.1.2.1 Oficina didática - AgryBirds	118
6.1.3 Circuito de aprendizagem Tinkercad um simulador Arduino	120
6.1.3.1 Oficina didática - Tinkercad e Arduino - Primeiros passos	120
6.1.3.2 Oficina didática - Acender um LED	122
6.1.4 Circuito de aprendizagem Arduino, o ambiente Editor de Códigos	125
6.1.4.1 Oficina didática - Instalação do Arduino	125
6.1.4.2 Oficina didática - Programando LED Blink	129
6.1.4.3 Oficina didática - LED com tempo programado na prática.	134
6.1.4.4 Oficina didática - Um LED com interruptor	135
6.1.4.5 Oficina didática - LED controlado com o acionamento de um botão	136
6.1.4.6 Oficina didática - 2 LEDs - Sinalizador de garagem	137
6.1.4.7 Oficina didática - Cinco LEDs simulando um semáforo	139
6.1.4.8 Oficina didática - Cinco LEDs piscando de forma alternada	141
6.1.4.9 Oficina didática - Potenciômetro controlando um LED	142
6.1.4.10 Oficina didática - Motor com potenciometro	145
6.1.4.11 Oficina didática - Sensor de Luminosidade para acender uma lâmpada	147
6.1.4.12 Oficina didática - Alarme de segurança com sensor infravermelho	149
6.1.4.13 Oficina didática - Robô carro 2 WD	152
6.1.5 Circuito de aprendizagem Aplicativos para celular	160
6.1.5.1 Oficina didática - App Inventor	160
6.1.5.2 Oficina didática - Aplicativo de mensagens via Bluetooth	164
6.1.5.3 Oficina didática - Aplicativo para controlar LEDs via Bluetooth	166

6.1.5.4 Oficina didática - Aplicativo para controlar motor DC via Bluetooth	178
6.2 Computação desplugada	182
6.2.1 Circuito de aprendizagem dinâmicas e atividades desplugadas	182
6.2.1.1 Oficina didática - Controle remoto	182
6.2.1.2 Oficina didática - Colorindo com números	184
6.2.1.3 Oficina didática - Passeio da tartaruga	186
<b>7. O Curso de formação de profissionais da educação - Formação Ensino Maker</b>	<b>189</b>
<b>8. Conclusão</b>	<b>193</b>
<b>9. Referências</b>	<b>195</b>



## 1. Introdução

Os professores de matemática, assim como os de outras áreas, ainda que são peritos em adaptar materiais diversos para o ensino, têm encontrado dificuldades em trabalhar conteúdos relacionados à robótica educacional e o Pensamento Computacional (PC) e, ainda, escolher metodologias que, de fato, despertam o interesse dos estudantes em compreender o mundo, as relações do cotidiano com os conteúdos e assim possibilitar o desenvolvimento de habilidades e competências necessárias para atuarem no mundo do trabalho e suas relações. Este trabalho, tem o objetivo de analisar o Currículo do estado do Espírito Santo (ES) e a Base Nacional Comum Curricular (BNCC) diante desse contexto, fazendo considerações relevantes com base em autores especialistas, para propor alternativas que podem auxiliar aos professores em sua prática dentro e fora de sala de aula, com a proposição de mediar ou nortear o profissional por meio de apropriação dos conceitos e a prática com proposta de circuitos de aprendizagem (planos de aula e/ou sequências didáticas), nos capítulos que seguem e também com projetos, experimentos e apresentação de ferramentas para seu desenvolvimento.

Alguns conceitos estão definidos neste trabalho com o objetivo de esclarecer e diferenciar uma das outras, visto que quando se pensa em pensamento computacional, programação, fluxograma, metodologia STEAM (Science, Technology, Engineering, Art e Math é uma nova abordagem de aprendizagem, que preza pela multidisciplinaridade em sala de aula) que são vários temas envolvendo tecnologias educacionais e que ainda são pouco utilizados ou, gera muitas dúvidas e confusões quanto às definições e conceitos. Nesse sentido, também objetiva-se mostrar a diferença entre a BNCC e o Currículo do ES, assim como apresentar possibilidades alinhadas aos novos hábitos oriundos da cibercultura, a qual é a cultura digital associada ao uso massivo e reflexivo das tecnologias digitais ao dia a dia do estudante e ao mundo do trabalho. Visto que, a cultura digital está associada ao uso massivo e reflexivo das tecnologias digitais, os estudantes são diretamente afetados pelos novos hábitos oriundos desta cibercultura.

Assim sendo, é preciso trazer para a realidade escolar esta nova maneira de pensar, agir e se relacionar com o meio em que vivem. É importante também, ressaltar que nossas escolas muitas vezes estão preocupadas em ensinar, mas, talvez o segredo esteja em como se aprende.

A distribuição dos capítulos neste trabalho foi organizado visando uma melhor compreensão conforme a necessidade gradual dos conhecimentos adquiridos, da forma a seguir:

- No capítulo **Pensamento Computacional, apropriação do conceito à prática**, temos a subdivisão feita em quatro subcapítulos.

Para tanto, são apresentados alguns dos principais fundamentos do Pensamento Computacional (PC) embasado em especialistas na temática. Podendo assim, compreender a necessidade do desenvolvimento das habilidades do Pensamento Computacional (PC) para as gerações da atualidade assim como para as gerações futuras.

→ *Pensamento Computacional (PC)*, que traz cinco outros subcapítulos no qual segue:

- Reflexões de alguns autores sobre o Pensamento Computacional;
- Pensamento Computacional e sua linha do tempo;
- Metodologias utilizadas para trabalhar o PC;
- Preciso aprender e ensinar os aprendizes a escreverem códigos para desenvolverem as habilidades do PC?;
- Pensamento Computacional na resolução de problemas.

→ *Computação Desplugada*;

→ *Exemplos de como a computação pode ser mais fácil de ensinar e aprender do que imaginamos*;

→ *Habilidades do profissional do futuro*.

- No capítulo **A (não) abordagem das tecnologias digitais na BNCC e no Currículo do Espírito Santo e o letramento digital**.

Neste capítulo, são apresentados conceitos sobre a BNCC, o Currículo do Espírito Santo e sobre o letramento digital e, como esses temas estão ou podem estar relacionados para assim fortalecer e possibilitar uma inclusão digital que de fato possa atender as necessidades da atualidade da vida escolar e profissional do estudante, assim como, apontar lacunas que devem ser preenchidas com planejamento adequado e eficaz pelos profissionais da educação. Vale acrescentar que, embora este trabalho trará apontamentos para o Currículo do Estado do Espírito Santo, não há o que impede que estes apontamentos possam se aplicar para os demais estados e seus currículos.

- *A BNCC e os necessários apontamentos para o desenvolvimento do PC;*
- *O Currículo do ES e os necessários apontamentos para o desenvolvimento do PC;*
- *Letramento e/ou educação digital.*

- No capítulo **Fluxogramas, algoritmos e linguagens de programação** é apresentado os conceitos básicos sobre cada um destes temas;

Portanto, é de suma importância que os estudantes possam ter conhecimentos mínimos necessários sobre fluxogramas, algoritmos e linguagens de programação, assim como a utilização destes para o ensino e aprendizagem da matemática, visto que muitas vezes o fluxograma aparece como objeto de conhecimento no Currículo do Espírito Santo e que deve ser apresentado como uma habilidade a ser desenvolvida pelos estudantes, sobre os conceitos de:

- *Fluxogramas;*
- *Utilização e aplicações de fluxogramas;*
- *Algoritmos;*
- *Linguagens de programação;*
- *Tipos de linguagens de programação;*
  - Linguagem em blocos;
  - Code.org (Angry Birds);
  - Scratch;
  - Pictoblox;
  - X-Logo;
  - HTML;

- Pascal;
- Linguagem C.

- No capítulo **Robótica Educacional, Arduino e componentes eletrônicos** é apresentado conceitos sobre cada um destes assuntos assim como a importância dos mesmos no desenvolvimento das habilidades relacionadas à competência 5 da BNCC.

Portanto, para poder compreender melhor sobre a Robótica educacional que embora seja um assunto muito falado, mas, ainda muito pouco ensinado em salas de aulas, principalmente quando esse assunto alinham-se a conteúdos de matemática e nesse sentido este capítulo é subdividido para apresentar cada um desses assuntos visando uma abordagem mais ampla dos assuntos:

→ *Robótica;*

→ *Robótica educacional;*

→ *Scratch;*

→ *Tinkercad;*

→ *Eletronica básica;*

- Componentes eletrônicos básicos para a robótica;
- Placas de circuitos eletrônicos;
- LED ou lampadas;
- Placa protoboard;
- Resistores;
- Potenciômetros;
- Motor de passo x motor DC;
- jumpers.

→ *MIT APP Inventor;*

→ *Arduino;*

- Modelos e tipos de arduino.

- No capítulo **Proposta didática - Circuitos de aprendizagem com oficinas didáticas plugadas e desplugadas**. Os circuitos foram escolhidos de forma que ao se trabalhar a sequência de oficinas didáticas propostas o estudante possa se apropriar dos conhecimentos mínimos necessários ao desenvolvimento das habilidades relacionadas.

Portanto, para que o professor possa se apropriar do assunto e assim planejar suas aulas de computação plugadas ou desplugadas foram disponibilizados seis circuitos de aprendizagem que se completam com as oficinas didáticas dentro de cada um deles como uma proposta de ensino e aprendizagem para aplicação nas aulas e desenvolvimento das habilidades do PC, conforme a organização seguinte:

→ *Computação plugada;*

- Circuito de aprendizagem X Logo;
- ❖ Oficina didática - Instalação do X Logo;
- ❖ Oficina didática - Desenhar um quadrado;
- ❖ Oficina didática - Desenhar um triângulo;
- ❖ Oficina didática - Desenhar um Retângulo;
- ❖ Oficina didática - Desenhar um Círculo usando procedimentos.
- Circuito de aprendizagem Code.org;
- ❖ Oficina didática - AngryBirds.
- Circuito de aprendizagem Tinkercad simulador Arduino;
- ❖ Oficina didática - Tinkercad e Arduino - primeiros passos;
- ❖ Oficina didática - Acender um LED.
- Circuito de aprendizagem Arduino, o ambiente Editor de códigos;
- ❖ Oficina didática - Instalação do Arduino;
- ❖ Oficina didática - Programando LED Blink;
- ❖ Oficina didática - LED com tempo programado na prática;
- ❖ Oficina didática - Um LED com interruptor;
- ❖ Oficina didática - LED controlado com o acionamento de um botão;
- ❖ Oficina didática - 2 LEDs - sinalizador de garagem;
- ❖ Oficina didática - Três LEDs simulando um semáforo;
- ❖ Oficina didática - Cinco LEDs piscando de forma alternada;
- ❖ Oficina didática - Potenciômetro controlando um LED;
- ❖ Oficina didática - Motor com potenciômetro;
- ❖ Oficina didática - Sensor de luminosidade para acender a lâmpada em um poste na rua;
- ❖ Oficina didática - Alarme de segurança - Sensor Infravermelho;

- ❖ Oficina didática - Robô carro 2 WD.
- Circuito de aprendizagem Aplicativos para celular;
- ❖ Oficina didática - APP Inventor;
- ❖ Oficina didática - Aplicativo de mensagens ou comandos via Bluetooth;
- ❖ Oficina didática - Aplicativo para controlar LED via Bluetooth;
- ❖ Oficina didática - Aplicativo para controlar motor DC via Bluetooth.

→ *Computação desplugada;*

- Circuito de aprendizagem dinâmicas e atividades desplugadas;
- ❖ Oficina didática - Controle remoto;
- ❖ Oficina didática - Colorindo com números;
- ❖ Oficina didática - Passeio da tartaruga.

- No capítulo **O curso de formação de profissionais da educação - Formação Ensino Maker**, é mostrado como ficou a formação chamada “Ensino Maker”<sup>1</sup>, em parceria com a equipe de formação da Secretaria de Educação do Estado do Espírito Santo.

Contudo, é apresentado através dos prints que foi retirado diretamente da plataforma de cursos, após a formação já formatada no ambiente virtual. Esse curso, foi pensado para ser disponibilizado em formato MOOC, na qual as atividades subsequentes ficam disponíveis automaticamente conforme a conclusão dos estudos e atividades de cada um dos tópicos, até a conclusão de todas as atividades propostas e a realização da avaliação final da formação.

- No capítulo **Conclusão** está apresentada uma súmula dos resultados obtidos, bem como uma análise abrangente sobre os mesmos, na qual entendemos pela superação das expectativas.

---

<sup>1</sup>Maker é uma palavra em inglês que significa fazedor, no sentido de criador, realizador, fabricante. O Maker pode compor com qualquer área de conhecimento.

Entende-se que o produto final deste trabalho contempla fortes características do modelo “Ensino Maker” a qual teve ótima aceitação pelos professores e público alvo e, para pesquisas futuras, ainda existem uma gama de possibilidades considerando que os temas PC e robótica educacional são amplos e a cada dia surgem mais novidades como por exemplo, o ChatGPT<sup>2</sup>, lançado em 30 de novembro de 2022.

---

<sup>2</sup>ChatGPT é um assistente virtual inteligente no formato chatbot online com inteligência artificial desenvolvido pela OpenAI, especializado em diálogo, lançado em novembro de 2022. O chatbot é um modelo de linguagem ajustado com técnicas de aprendizado supervisionado e por reforço. Disponível em: Wikipédia. Acesso em 06/2023.

## 2. Pensamento Computacional, apropriação do conceito à prática

Este capítulo apresenta alguns dos principais conceitos sobre **Pensamento Computacional (PC)** segundo perspectiva encontrada na literatura científica, além de mostrar algumas das possibilidades de apropriação no que diz respeito ao tema proposto. Assim, pretende-se apresentar pontos fundamentais da temática e ainda, fazer apontamentos para metodologias possíveis a aplicar no desenvolvimento das habilidades e competências relacionadas.

### 2.1 Pensamento Computacional (PC)

Segundo Steve Jobs (cofundador da Apple Inc.), “todas as pessoas deveriam aprender a programar computadores, porque isso ensina a pensar”, e de fato, para o desenvolvimento das competências<sup>3</sup> do Pensamento Computacional, isso tem se tornado tão importante quanto saber ler e escrever. Outro apontamento importante nesse sentido foi dado pelo Fórum Econômico Mundial ([WEF](#)), o qual elencou, dentre as dez habilidades propostas que preparam o estudante para a vida e para as profissões do futuro, quatro habilidades do Pensamento Computacional como essenciais no desenvolvimento pessoal e profissional.

Além disso, o desenvolvimento das **habilidades do PC**<sup>4</sup> auxilia no desenvolvimento de tarefas cotidianas do indivíduo, portanto, Wing (2016) acredita que o **Pensamento Computacional é uma habilidade fundamental para todos**, sem exceção, acrescentando ainda que o PC são justamente os processos de pensamento envolvidos na formulação de um problema e que expressam sua solução ou soluções eficazmente, de tal forma que uma máquina ou uma pessoa possa realizar. Envolvendo assim, aplicação de aspectos da computação em diferentes áreas da vida, propondo soluções criativas de forma crítica e estratégica a problemas e soluções.

---

<sup>3</sup> As Competências do Pensamento Computacional são: abstração, decomposição de problemas em partes, reconhecimento de padrões e análise e representação de dados.

<sup>4</sup> Algumas das habilidades do Pensamento Computacional são: a autonomia, a alfabetização digital e a construção do pensamento lógico.



### 2.1.1 Reflexões de alguns autores sobre o Pensamento Computacional

A alfabetização e seu processo de ensino-aprendizagem contempla não apenas a leitura, a escrita e a aritmética, mas também o desenvolvimento de habilidades e competências da esfera do Pensamento Computacional, da sustentabilidade e da responsabilidade ambiental, por exemplo.

Nesse sentido, o “ensino de computação na educação básica é um assunto que vem sendo discutido e pesquisado na academia” (BRACKMANN, 2017; FRANÇA *et al.*, 2014; ANDRADE *et al.*, 2013), estando relacionado ao conceito de Pensamento Computacional, o qual é defendido como um conjunto de habilidades que permite a resolução de problemas de forma otimizada, cujo desenvolvimento é tão importante para os estudantes no processo da aprendizagem quanto o ensino da matemática, das ciências e/ou das linguagens. Contudo, apesar de o Pensamento Computacional ser um requisito elementar na formação de qualquer estudante, ele “ainda é pouco abordado no contexto escolar brasileiro, estando restrito apenas a projetos e atividades extracurriculares” (BARCELOS; SILVEIRA, 2012). Com essa mesma perspectiva, Brackmann afirma que:

“O **Pensamento Computacional** é uma abordagem de ensino que usa diversas técnicas oriundas da Ciência da Computação e vem gerando um novo foco educacional no quesito inovação nas escolas mundiais como um conjunto de competências de solução de problemas que devem ser compreendidos por uma nova geração de estudantes em conjunto com as novas competências do século 21 (i.e., pensamento crítico, colaboração, etc.).” (BRACKMANN, 2017, p. 8).

Nesse processo, de desenvolvimento das habilidades e conceitos do PC para o ensino das ciências e da matemática, geralmente utilizamos modelos e/ou situações problemas que facilitam o ensino e a aprendizagem de conteúdos, e também a compreensão e o entendimento da teoria relacionada à prática. Nessa mesma direção, Alencar traz que “o desenvolvimento do Pensamento Computacional parece estar diretamente ligado à capacidade de visualizar e interpretar problemas para então agir sobre eles” (ALENCAR, 2020, p.111).

Outra definição tão importante quanto a anterior é a de Bocconi, a qual afirma que o “Pensamento Computacional (PC) é um processo de pensamento (ou uma habilidade de pensamento humano) que usa abordagens analíticas e algorítmicas para formular, analisar e resolver problemas” (BOCCONI, 2016, p. 9).

Portanto, a resolução de problemas no ensino da matemática e das ciências, sejam estes reais ou imaginários, são formas ou métodos desenvolvidos e utilizados para que os professores consigam alcançar o objetivo de ensinar a seus aprendizes a compreenderem situações da realidade, assim como a pensarem em situações e possibilidades, e visualizarem de forma eficiente as possíveis soluções para cada caso, considerando e relacionando com o desenvolvimento e aplicação das habilidades e competências do pensamento computacional, de forma isolada ou em sua generalização.

### 2.1.2 Pensamento Computacional (PC) e sua linha do tempo

O termo "Computational Thinking" (Pensamento Computacional) foi apresentado pela primeira vez em 1980 por Seymour Papert e se refere a um conjunto de habilidades desenvolvidas em ciência da computação que podem ser aplicadas para solucionar problemas em diversas áreas de conhecimento, promovendo assim a interdisciplinaridade. Algum tempo depois, esse termo foi introduzido por Jeannete Wing em seu artigo *Computational Thinking - Communications*, no qual a autora afirma ser essa uma habilidade fundamental para todos, e não somente para os cientistas da computação.

A geração zero é datada de 3500 a.C. (antes de Cristo) com o surgimento do ábaco e em seguida com as máquinas mecânicas de calcular e se estende até 1945, quando se inicia a primeira geração da computação com o surgimento das máquinas de calcular com válvulas, e a partir de 1955 iniciou-se a segunda geração com a evolução e uso dos transistores, essa segunda geração termina com aparecimento dos circuitos integrados em 1965 até 1980 quando se data o final da terceira geração e surge a quarta geração com os circuitos integrados em larga escala de 1980 a 1995, já a quinta geração da evolução da computação, Figura 1, é a qual estamos vivenciando com a chamada computação invisível (internet, computação em nuvens, tecnologia Bluetooth, TV digital, computação vestível, etc.).

Figura 1 - Evolução da computação



Fonte: Página computacional.com<sup>5</sup>

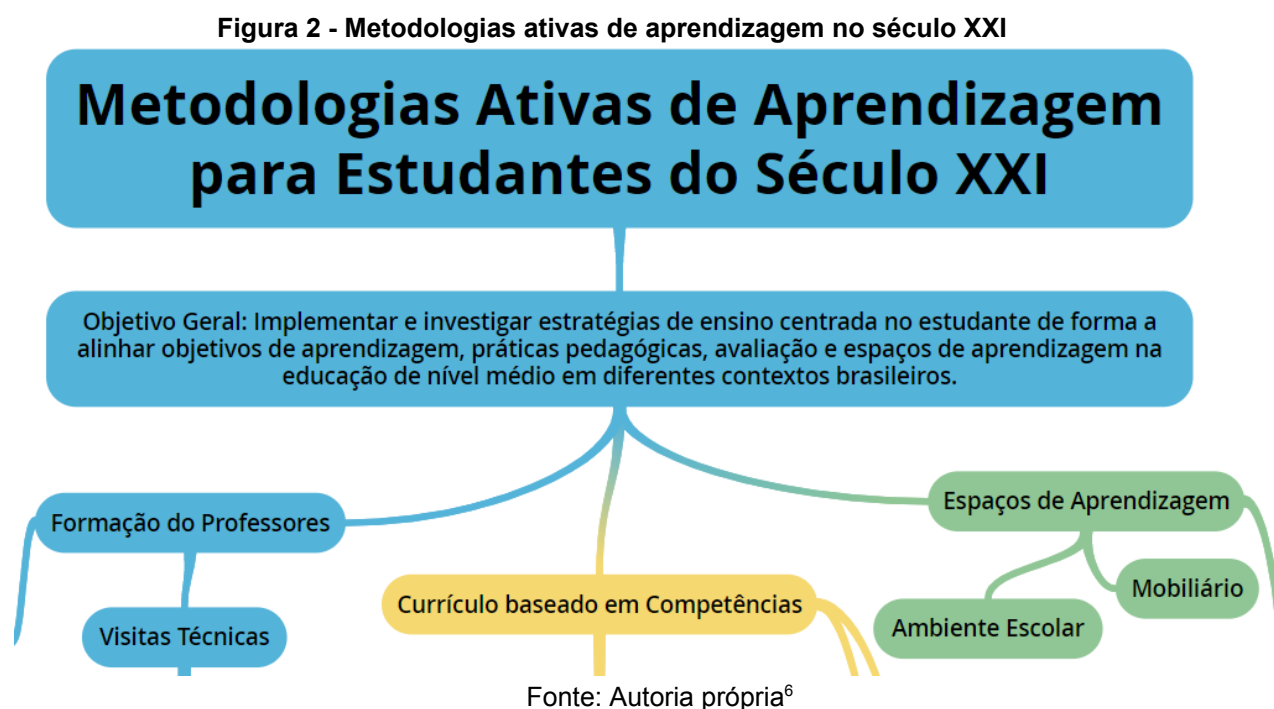
### 2.1.3 Metodologias utilizadas para trabalhar o PC

As metodologias ativas permitem aos estudantes expandirem suas ideias, estimulando o processo de inovação e criatividade, portanto, as mais indicadas para se trabalhar o desenvolvimento do pensamento computacional, “a combinação do pensamento crítico com os fundamentos da computação define uma metodologia para resolver problemas” (Wing, 2006), o estudante se torna protagonista do seu desenvolvimento enquanto o professor tem uma importância fundamental como mediador do conhecimento, na proposta do construcionismo, a qual é baseada na realização de uma ação concreta que resulta em um produto palpável, desenvolvido com o concurso do computador e de interesse de quem o produz. Assim, as informações são implementadas no computador e passadas ao estudante na forma de tutorial, exercício, prática ou jogo, sempre de forma mediada e com o educando como protagonista da aprendizagem. Dessa maneira, o professor por sua vez, assume um papel de facilitador, guiando os estudantes em seus processos de aprendizagem e este passa a ser compartilhado. Conforme Bacich e Moran:

<sup>5</sup> Computacional.com. Disponível em: <<https://www.computacional.com.br/atividades/evolucaodacomputacao.pdf>>. Acesso em 03/2022.

“O fato de elas serem ativas está relacionado com a realização de práticas pedagógicas para envolver os alunos, engajá-los em atividades práticas nas quais eles sejam protagonistas da sua aprendizagem. Assim, as metodologias ativas procuram criar situações de aprendizagem nas quais os aprendizes possam fazer coisas, pensar e conceituar o que fazem e construir conhecimentos sobre os conteúdos envolvidos nas atividades que realizam, bem como desenvolver a capacidade crítica, refletir sobre as práticas realizadas, dar e receber feedback, aprender a interagir com colegas e professor, além de explorar atitudes e valores pessoais.” (BACICH E MORAN, 2018, p. 81).

Contudo, é importante o papel do professor nessa construção e no desenvolvimento do aprendizado do estudante, assim como compreender como funciona esta metodologia. Sendo assim, na Figura 2, apresentamos um mapa mental das metodologias ativas para melhor entendimento do tema:



Existem diversos tipos de metodologias ativas e cada uma delas tem um potencial que deve ser analisado de acordo com a necessidade, para o desenvolvimento das habilidades e competências conforme a finalidade e planejamento do profissional.

<sup>6</sup> Metodologias ativas de aprendizagem no século XXI. Disponível em: <<https://www.goconqr.com/pt-BR/mindmap/36519645/metodologias-ativas-de-aprendizagem-para-estudantes-do-seculo-xxi>>. Criada em 05/2022.

Essa transformação dos modelos de ensino tradicionais com o uso de metodologias ativas conta com o auxílio de tecnologias digitais, computadores, tablets, celulares, entre outros dispositivos que passam a ser ferramentas necessárias nesse processo, facilitando o acesso a informações e recursos.

Sendo assim, é válido listar algumas dessas metodologias ativas, o objetivo aqui não é explicar sobre cada uma delas apenas apontá-las:

- Aprendizagem baseada em problemas;**
- Ensino híbrido;
- Sala de aula invertida;
- Aprendizagem baseada em projetos;**
- Aprendizado entre pares;
- Gamificação;
- Rotação por estações de aprendizagem;
- Cultura Maker;**
- Estudos do meio;
- Estudos de casos;
- Storytelling;
- Design thinking;
- Outras.

É destacado pelo menos três delas: Cultura Maker; Aprendizagem baseada em problemas; Aprendizagem baseada em projetos, consideradas as mais usuais quando pretende-se trabalhar a robótica educacional. É importante também, destacarmos que nesse processo, um bom planejamento faz toda diferença para um trabalho eficiente com resultados satisfatórios e este, pode ser um diferencial, haja vista que há grandes vantagens mas também, diversos desafios para a aplicação das metodologias ativas:

- Falta de tempo e/ou de conhecimento para elaboração de planos de aulas com metodologias ativas;
- Falta de acesso a tecnologias, dispositivos e conexão para professores e/ou estudantes;

- O uso da tecnologia como um recurso restrito ao professor e não ao aluno, o que interfere na autonomia deste diante do conteúdo a ser aprendido;
- Resistência na transformação da cultura do ensino, atrelada a noções tradicionais de ensino-aprendizagem;
- Paralisar a transformação na simples ação de incluir equipamentos e dispositivos na sala de aula, sem alterar o planejamento e o formato das aulas. Desse modo, os estudantes não observam uma mudança significativa no seu processo de ensino-aprendizagem e não se tornam figuras centrais dele;
- Resistência do corpo docente e/ou de outros setores para aplicação das metodologias, uma vez que elas representam uma mudança profunda no ensino tradicional;
- Entre outras.

Embora muitos sejam os desafios, os benefícios das metodologias ativas na postura e no comportamento dos estudantes, tornam-se visíveis. O protagonismo transforma e promove uma guinada essencial, para que os estudantes se sintam verdadeiramente parte da construção do seu conhecimento, tendo autonomia e capacidade de autogestão, despertando a proatividade, o dinamismo, a criatividade, o pensamento crítico, a comunicação, o trabalho em equipe e a busca por novos desafios na solução de problemas.

#### 2.1.4 Preciso aprender e ensinar os aprendizes a escreverem códigos para desenvolver as habilidades do PC?

Para se aprender ou ensinar o PC, mais especificamente desenvolver as habilidades do PC, as quais vão além de saber navegar na web, publicar páginas ou postagens nas redes sociais e fazer uso das diversas ferramentas digitais, **não necessariamente é preciso escrever códigos ou algoritmos de computadores**. Abstração e algoritmos são duas bases do PC que permitem aos estudantes, por meio do desenvolvimento e da construção do pensamento lógico e da autonomia serem produtores de recursos digitais, avançando assim na construção de soluções para problemas comuns de forma inovadora, com rapidez e eficiência.

Isso porque o uso das ferramentas digitais, que são práticas comuns aos nativos digitais<sup>7</sup>, são habilidades que o indivíduo desenvolve ao longo de sua trajetória de vida, **não sendo necessariamente preciso se tornar um expert na ciência da computação para tal.**

No entanto, é preciso sim um estímulo para que os estudantes sejam crianças, adolescentes, jovens e/ou adultos que possam emergir nessa busca pelo seu desenvolvimento pleno, para que assim desenvolvam seus próprios meios de solucionar ou criar problemas, seguindo as **etapas do pensamento computacional**, as quais adéquam-se às situações simples ou complexas do dia a dia do indivíduo, no entanto, lembrando sempre que a maioria dos elementos do PC são processos e não conteúdos.

Essas etapas, ou competências do PC são: abstração, decomposição de problemas em partes, reconhecimento de padrões e análise e representação de dados. A seguir, na Figura 3, temos algumas das características e ações destas etapas.



Fonte: Autoria própria.

<sup>7</sup> Nativos digitais segundo o educador e pesquisador Marc Prensky (2001) é a geração de jovens nascidos a partir da disponibilidade de informações rápidas e acessíveis na grande rede de computadores – a Web.

### 2.1.5 Pensamento computacional na resolução de problemas

O pensamento computacional não depende de computadores e nem de linguagem computacional, antes busca entender como o funcionamento do computador pode ser útil como um instrumento que ajuda a aumentar o poder cognitivo dos estudantes, considerando suas restrições de recursos, a reformulação, a transformação e, quando couber, a incorporação do problema. Isto é, é um processamento paralelo ao pensar recursivamente<sup>8</sup>. Segundo Brackmann:

“O Pensamento Computacional é uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas colaborativamente através de passos claros de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente.”  
(BRACKMANN, 2017, p. 29).

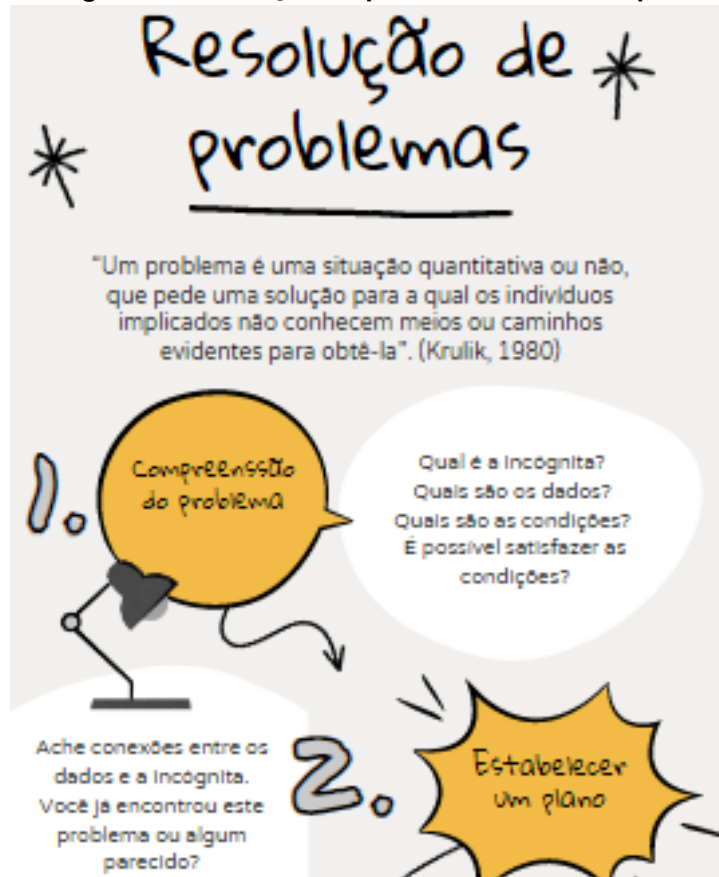
Uma imediata convergência entre essa definição do pensamento computacional e as competências e habilidades relacionadas à matemática na analítica da educação, as quais vão além da leitura, da escrita e da aritmética, se dá por meio da resolução de problemas, como na proposta de Polya, que em seu trabalho aponta como “fundamental compreender, planejar, executar, verificar e interpretar o problema para que a partir da abstração (definida como a combinação de analogia, generalização e especialização) e a decomposição da situação-problema na busca com sucesso na resolução” (POLYA, 2004, p. 17). Nesse sentido, foi pensado um esquema, Figura 4, para facilitar o entendimento desse modelo de resolução de problemas, como segue:

---

<sup>8</sup> 'Recursividade' é um termo usado de maneira mais geral para descrever o processo de repetição de um processo ou método similar ao que já fora apresentado.



Figura 4 - Resolução de problemas e suas etapas

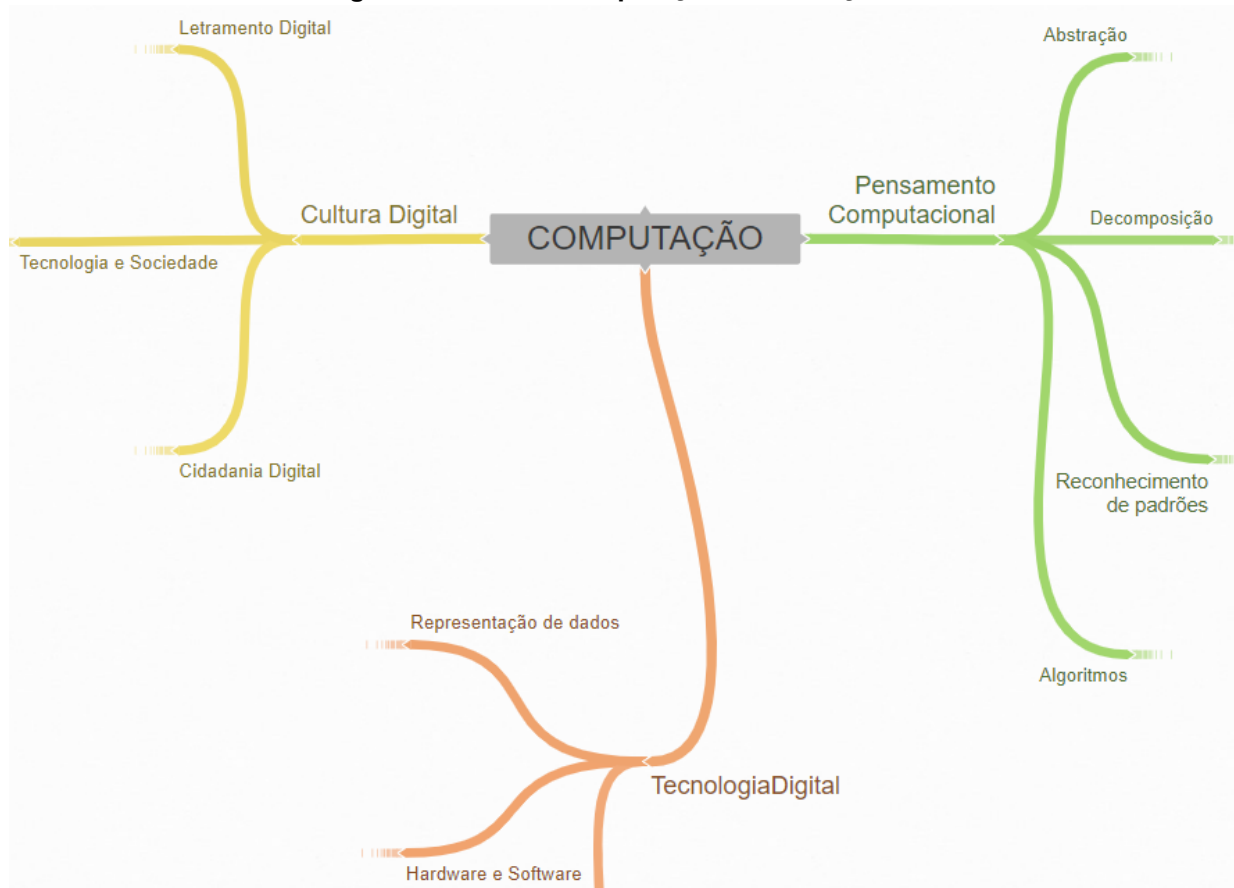


Fonte: Autoria própria<sup>9</sup>

No entanto, não existe uma fórmula pronta. Isto é, busca-se resolver situações-problema do dia a dia a partir de fundamentos da computação, de forma prática, fácil e lúdica. **Assim, um simples manual prático, com fluxogramas, pode facilitar o entendimento dos conceitos de algoritmos trabalhados na computação.** A propósito, os fluxogramas são utilizados na escrita de algoritmos para facilitar o entendimento e proporcionar uma visão mais ampla da situação problema, dos passos a seguir para solucionar problemas. Podemos dizer ainda que nesta metodologia, embora seja possível oferecer aos estudantes uma receita pronta, como uma receita de bolo, também é possível que estes desenvolvam seus próprios meios de criar e também de solucionar uma determinada situação problema. A imagem a seguir, Figura 5, ilustra a computação associada ao PC, à cultura digital e a tecnologia digital e suas interconexões.

<sup>9</sup> Resolução de problemas e suas etapas. Disponível em: <<https://acesse.one/resolucaodeproblemasPolya1945-Kruulik-Rudnick1980>>. Acesso em 05/2022.

Figura 5- Fluxo da computação na educação



Fonte: Autoria própria<sup>10</sup>

Nesta perspectiva, os computadores e seus softwares nos ajudam a solucionar problemas cotidianos que nenhum de nós seria capaz de enfrentar sozinho, sem os métodos e modelos computacionais, e é assim, baseado no poder e nos limites dos processos computacionais executados por máquinas ou por humanos, mostrando assim que o PC confronta o enigma da inteligência da máquina. De fato, a tecnologia mostra o seu potencial para a educação como um campo de possibilidades para testes, manipulação e experimentação, tendo como fundamentação a relevância dos computadores para a sociedade, o que torna necessário o fomento do desenvolvimento de habilidades relacionadas à programação de computadores. Segundo Brackmann:

<sup>10</sup>Fluxo da Computação na Educação, disponível em: <<https://coggle.it/diagram/YnLlnA8o3JWkdPly/t/computa%C3%A7%C3%A3o/9e1f41059550bbe62a04f2747427a32702dfcab4dc2ea011fdf04fa305f9403b>>. Acesso em 05/2022.

“O Pensamento Computacional é uma abordagem de ensino que usa diversas técnicas oriundas da Ciência da Computação e vem gerando um novo foco educacional no quesito inovação nas escolas mundiais como um conjunto de competências de solução de problemas que devem ser compreendidos por uma nova geração de estudantes em conjunto com as novas competências do século 21 (i.e., pensamento crítico, colaboração, etc.)” (BRACKMANN, 2017, p. 8).

Porém, no que tange ao processo de ensino/aprendizagem de programação, a literatura da área tem evidenciado que os docentes, ao ensinar programação, têm obtido menos êxito do que deveriam e precisariam ter (McGettrick *et al.*, 2005; Qian & Lehman, 2017; Robins, 2019; Vihavainen *et al.*, 2014; Watson & Li, 2014).

Isto se deve, talvez, ao fato de que, na atualidade, com o advento das novas tecnologias, o professor já não é mais o detentor do conhecimento e os modelos tradicionais de educação já não são efetivos, visto que as informações e as mudanças acontecem de forma rápida e os meios de comunicação são mais eficazes em transmitir informações. Nesse sentido, cabe ao professor se reinventar, adaptando-se às novas metodologias, reaprendendo novas formas de ensinar e aprender, o professor, dessa forma, tornar-se um mediador do conhecimento no ambiente escolar. Esse raciocínio é corroborado ainda por Polya (2006), quando este autor afirma que “um dos mais importantes deveres do professor é o de auxiliar os seus alunos, o que não é fácil, pois exige tempo, prática, dedicação e princípios firmes” (POLYA, 2006, p.1).

Além disso, as transformações atuais desencadeadas a partir das Tecnologias Digitais de Informação e Comunicação (TDICs) interferem na subjetividade dos sujeitos e em seus modos de viver, de se relacionar e também de aprender, sendo que “a melhor aprendizagem ocorre quando o aprendiz assume o comando” (PAPERT, 1994, p. 37). Com isso, é urgente a articulação com as atuais grades curriculares para formação e potencialização do uso das ferramentas tecnológicas na prática, além da construção de novas formas de se pensar a formação de professores a partir da integração entre a experimentação e a investigação. Porém, para tanto, é preciso formação de professores voltada para o uso efetivo da tecnologia e seus recursos integrados à experimentação e investigação, e que de fato seja eficiente na quebra de alguns paradigmas.

As Tecnologias Digitais de Informação e Comunicação (TDICs) citadas acima são entendidas como um conjunto de recursos tecnológicos integrados que proporciona, por meio das funções de software e telecomunicações, a automação e a comunicação dos processos de negócios, das pesquisas científicas e do processo de ensino-aprendizagem (ROCHA *et al.*, 2019). Assim, podemos entender que, embora as tecnologias não solucionem todos os problemas da educação, por meio delas, se usadas de maneira correta, (como ferramenta de ensino e aprendizagem, não apenas como apetrecho) teremos mudanças significativas e positivas:

“As tecnologias novas não poderiam ser indiferentes a nenhum professor, por modificarem as maneiras de viver, de se divertir, de se informar, de trabalhar e de pensar. Tal evolução afeta, portanto, as situações que os alunos enfrentam e enfrentarão, nas quais eles pretensamente mobilizam e mobilizarão o que aprendem na escola.” (PERRENOUD, 2000, p.138).

Acredita-se que “deve haver consenso em dominar as técnicas necessárias à prática do uso da tecnologia e ao conhecimento pedagógico” (VALENTE, 2005, p.20), portanto, pode-se dizer que esse é um bom motivo pelo qual todos devemos estudar PC. Nesse sentido, a Base Nacional Comum Curricular (BNCC), publicada em 2018, traz um pouco do PC enquanto tema transversal, propondo nos itinerários formativos do Ensino Médio que é possível trabalhar de forma mais incisiva o ensino e aprendizagem da competência 5 da BNCC, a qual traz um conjunto de habilidades que vão permitir intensificar, identificar e resolver problemas a partir de ensinamentos da Ciência da Computação<sup>11</sup>:

“Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.” (BNCC, 2018).

---

<sup>11</sup> Ciência da Computação é o ramo que soluciona diversos problemas por meio da tecnologia de computadores.

Contudo, criar tecnologias na atualidade já não é uma tarefa apenas de profissionais da área de TI - Tecnologia da Informação, visto que mesmo que as escolas ainda precisem intensificar o ensino e aprendizagem e o desenvolvimento dessas habilidades da computação, já são muitas as soluções de problemas cotidianos propostas por jovens aprendizes, os quais apresentam iniciativas incríveis e motivadoras. Como exemplo, temos o carioca Theo Correia que, com apenas 10 anos, desenvolveu um game para conscientização e preservação do meio ambiente, o Ciclovias Verdes, inspirado na cidade de Niterói (RJ), e já pensando em empreendedorismo o jovem David Braga, de Alagoas, que aos 14 anos criou um aplicativo para vender materiais escolares pela internet.

Nesse sentido, os estudantes vêm se desenvolvendo, criando soluções que auxiliam a solucionar problemas cotidianos, com a ajuda da escola e de mediadores, pode-se dizer que muitas escolas estão avançando nesta área de computação e programação, na estruturação de salas maker, e no uso de metodologias ativas e ensino maker, assim como educação STEAM. Porém, isso precisa fazer parte do currículo da escola, afinal, é importante que os estudantes dominem os conceitos e as habilidades que essa área traz e que esse conhecimento seja útil na prática, de modo que resolvam problemas e que tragam soluções para a sua vida e de sua comunidade, uma vez que, para se exercer cidadania na atualidade, o indivíduo tem que conhecer soluções computacionais que estão presentes no seu dia a dia.

Tucker (2003, p. 5) afirma que “é necessário que nos dias atuais cada cidadão compreenda pelo menos os princípios básicos da Ciência da Computação”, visto que “a Ciência da Computação nesta história não é mais vista como a ajudante da engenharia de software, mas como parte da agenda científica: produzir novas ferramentas para mineração de dados, visualização, programação, etc. que permitam novos tipos de oportunidades de pesquisa” (BUNDY, 2007, p.1), ou seja, é uma necessidade para a humanidade, no contexto do desenvolvimento sociocultural, econômico e sustentável. Porém, como nos currículos de Ensino Básico regular no Brasil não há disciplinas específicas que incitem o pensamento e o raciocínio lógico computacional, a aprendizagem da lógica computacional pode ser morosa nesse contexto. Com a mesma linha de raciocínio, Coll (2018) escreve que:

“O pensamento computacional é o expoente de uma cultura epistemológica ligada ao desenvolvimento e generalização das mídias e redes digitais [que] surgiram com a computação pessoal, a internet, e continuará com IA e possivelmente blockchain, entre outros meios” (COLL, 2018, p.3).

A despeito de, ainda é possível constatar que muitos autores vão além e nos mostram que o desenvolvimento das habilidades do PC está ligado a uma nova forma de pensar, de abstrair e de resolver situações problemas. Nesse sentido, Coll (2018) continua:

“tampouco sua vinculação com uma nova forma de alfabetização, a alfabetização digital, e com a necessidade que surge das pessoas no cenário atual de adquirir e desenvolver uma série de habilidades básicas para poder crescer e se desenvolver adequadamente no âmbito da cultura digital.” (COLL, 2018, p.3).

Assim, o letramento digital e o desenvolvimento das habilidades de resolução de problemas simples e complexos é uma necessidade para a compreensão do mundo contemporâneo. Nesse sentido, Brackmann traz que:

“O Pensamento Computacional envolve identificar um problema complexo e quebrá-lo em pedaços menores e mais fáceis de gerenciar (DECOMPOSIÇÃO). Cada um desses problemas menores pode ser analisado individualmente com maior profundidade, identificando problemas parecidos que já foram solucionados anteriormente (RECONHECIMENTO DE PADRÕES), focando apenas nos detalhes que são importantes, enquanto informações irrelevantes são ignoradas (ABSTRAÇÃO). Por último, passos ou regras simples podem ser criados para resolver cada um dos subproblemas encontrados (ALGORITMOS).” (BRACKMANN, 2017, p. 33).

Vale ressaltar aqui que estes quatro pilares do PC trazidos por Brackmann são utilizados para atingir o seu objetivo principal que é a resolução de problemas, o que é fundamental para os estudantes nas diversas áreas de conhecimento, em especial na matemática.

“A importância dada à Resolução de Problemas é recente e somente nas últimas décadas é que os educadores matemáticos passaram a aceitar a ideia de que o desenvolvimento da capacidade de se resolver problemas merecia mais atenção. [...] Hoje, a tendência é caracterizar esse trabalho considerando os estudantes como participantes ativos, os problemas como instrumentos precisos e bem definidos e a atividade na resolução de problemas como uma coordenação complexa simultânea de vários níveis de atividade.” (ONUChic, p. 203, 1999).

Assim, a resolução de problemas pode ser utilizada como um método para aprender a desenvolver habilidades e competências cotidianas do estudante, na busca de melhorias do território e do trabalho onde permeiam.

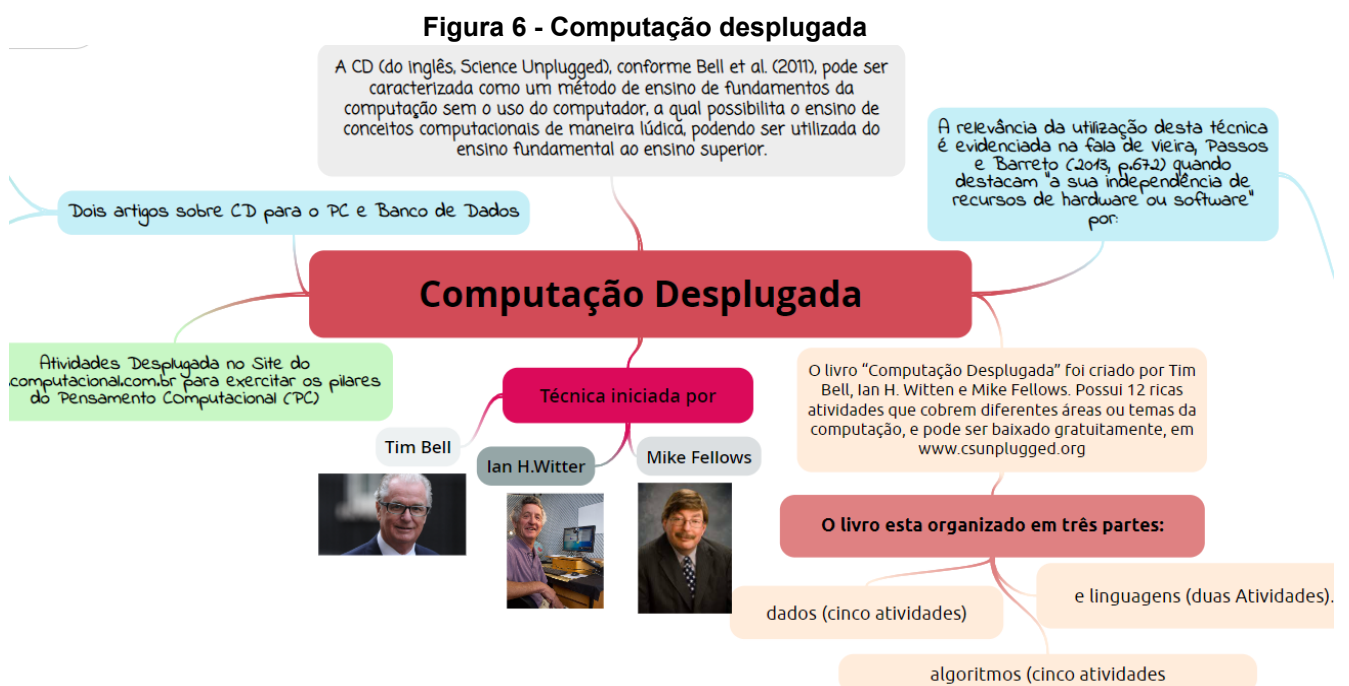
## **2.2 Computação desplugada**

A **computação desplugada** é um método voltado para o ensino-aprendizado de conceitos computacionais que dispensa a utilização de computadores, tablets, smartphones e similares (BELL *et al.*, 2011), e que pode ser aplicado em localidades remotas e/ou com infraestruturas deficitárias, como é o caso de muitas escolas públicas brasileiras. Assim, embora seja comum entre nós professores, pensarmos que é sempre necessário o uso do computador para se trabalhar computação e para se desenvolver habilidades do Pensamento Computacional, Bell nos garante que a computação desplugada tem justamente “o objetivo de ensinar os fundamentos sobre a Ciência da Computação sem o uso do computador” (BELL *et al.*, 2011, p. 3) e em seu livro apresenta exemplos e formas de como isso é possível de ser realizado. A proposta envolve projetos de sistemas, pensamento recursivo, abstração e decomposição para resolver problemas simples e complexos, além do entendimento do comportamento humano, da modularização, do uso de heurísticas de raciocínio, etc.

As atividades da computação desplugada ajuda aos estudantes a compreenderem como descrever um problema, identificar quais são os detalhes importantes a que eles precisam se atentar para resolver esse problema, dividi-lo em pequenos passos lógicos para que eles possam criar um processo de resolução, avaliar esse processo e sua aplicação de forma efetiva, e isto remete ao ensino-aprendizagem do PC e o desenvolvimento de suas habilidades.

Portanto, o profissional que se dispõe a aplicar, ensinar ou até mesmo pesquisar sobre o tema PC pode criar atividades, ou pesquisar/adaptar atividades de outros pesquisadores que já desenvolveram experiências desplugadas, sendo importante para esse profissional apropriar-se de metodologias contemporâneas de ensino-aprendizagem para repassar com segurança os conteúdos de computação e, assim, de fato, estimular a abstração e o pensamento lógico-matemático do público-alvo, que são os estudantes do Ensino Básico à especialização e até mesmo os profissionais atuantes nas diversas profissões que estão evoluindo para agregar tecnologia e se tornando cada vez mais multidisciplinares, o que vem intensificando os impactos positivos do emprego das tecnologias nas diversas áreas e setores.

Para saber mais sobre computação desplugada, observe o esquema na Figura 6, a seguir:



Fonte: Autoria própria<sup>12</sup>.

<sup>12</sup> Computação Desplugada. Disponível em: <<https://www.goconqr.com/pt-BR/mindmap/36519840/computacao-desplugada>>. Acesso em 05/2022.



### **2.3 Exemplos de como a computação pode ser mais fácil de ensinar e aprender do que imaginamos**

O uso do computador enquanto máquina de ensinar consiste apenas na informatização dos métodos de ensino tradicionais (Valente, 1993, p.32), por isso, em muitos casos, o computador é usado apenas para informatizar os processos de ensino já existentes, não como fonte de conteúdo a ser de fato estudado e ensinado, o que nos coloca em posição de refletir sobre as formas de se relacionar tecnologia e educação e sobre como aquela pode nos auxiliar, quando aplicada da forma correta para o ensino e aprendizagem eficaz.

Neste sentido, existem várias iniciativas e/ou projetos visando disseminar questões relacionadas ao universo do PC e das experiências educativas com o ensino da computação entre educadores, e assim aproximar crianças, adolescentes, jovens e adultos desse universo. Dentre essas iniciativas, podemos citar o Programaê, da Fundação Telefônica Vivo, o CODE.ORG, a Hora do código e o Instituto Ayrton Senna, além de Institutos Federais, Universidades, escolas particulares e outras instituições que estão dando maior ênfase ao assunto por exigência do mercado e da atualização do currículo.

Além disso, vários outros projetos têm surgido com objetivo de estimular o desenvolvimento do PC, a exemplo do Computational Thinking in K-12 Education Leadership Toolkit, um conjunto de ferramentas desenvolvido por líderes de diferentes escolas, em conjunto com a Computer Science Teachers Association (CSTA), a International Society for Technology in Education (ISTE) e a National Science Foundation (NSF) [CSTA *et al.* 2010].

Nessa mesma perspectiva, temos a Code.org, Figura 7, uma organização sem fins lucrativos dedicada à expansão do acesso à Ciência da Computação em escolas que oferecem cursos gratuitos e oportunidades a estudantes do mundo todo, com objetivo de desenvolvimento do PC, além de disponibilizar em seu site um estúdio prático e de fácil adaptação aos estudantes que dominam ou não a programação.

Figura 7 - Estudio Code.org



Fonte: Página Code.org<sup>13</sup>

**Para saber mais sobre a Code.org, acesse:**

Site: <https://code.org/international/about>

Projetos: <https://studio.code.org/projects/public>

Parceria Fundação Telefonica Vivo: <https://programae.org.br/hora-do-codigo/>

Por fim, outra indicação é a ferramenta online e gratuita **Tinkercad**, Figura 8, uma plataforma na qual há vários modelos e exemplos por meio dos quais também é possível desenvolver as habilidades do PC de forma fácil e intuitiva. Por meio dessa ferramenta é possível ainda realizar uma simulação de circuitos elétricos analógicos e digitais com imagens em 3D que facilitam a visualização via método geometria sólida construtiva, gerando assim objetos mais complexos em visualizações aproximadas da realidade.

<sup>13</sup> Code.org. Disponível em: <<https://studio.code.org/hoc/1>>. Acesso em 2022.

Figura 8 - Tinkercad

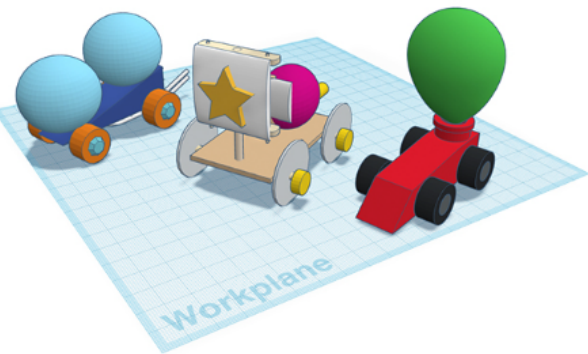


Da mente ao projeto em minutos

O Tinkercad é um aplicativo Web gratuito e fácil de usar que fornece a projetistas e engenheiros a próxima geração de habilidades fundamentais para inovação: projeto 3D, eletrônica e codificação.

Iniciar edição

Entrar na sua a...



Projete um carro acionado por balão Experimente! < || >

Fonte: Página Tinkercad<sup>14</sup>

**Para saber mais sobre a ferramenta Tinkercad, acesse:**

Site: <https://www.tinkercad.com/>

Projetos: <https://www.tinkercad.com/things>

## 2.4 Habilidades do profissional do futuro

A necessidade de adquirir habilidades e competências para o uso e domínio das ferramentas tecnológicas atuais, tanto no cotidiano do mundo do trabalho quanto no mundo digital, tem feito com que as pessoas busquem cada vez mais informações e cursos com objetivo de dominar tais habilidades. Assim, pode-se observar que o PC está ligado ao digital e, conseqüentemente, ao conhecimento, domínio e uso de ferramentas e dispositivos digitais.

Nesse sentido, embora muitos não possuam habilidades para lidar com as ferramentas tecnológicas do cotidiano, em algum momento já tiveram a necessidade de utilizá-las, seja em um aplicativo de celular, em uma página de internet ou em um totem de uma agência bancária, o que nos mostra o quão é importante trabalhar na Educação Básica com o desenvolvimento das habilidades do PC para o uso adequado e eficiente dessas ferramentas.

<sup>14</sup> Tinkercad. Disponível em: <<https://www.tinkercad.com/>>. Acesso em 2022.

Além disso, o Fórum Econômico Mundial, WEF<sup>15</sup>, apontou que são 10 as habilidades essenciais aos profissionais do futuro, quais sejam, criatividade, pensamento crítico, flexibilidade cognitiva, resolução de problemas complexos, inteligência emocional, capacidade de julgamento e tomada de decisões, gestão de pessoas, coordenação, orientação para servir e negociação, sendo que 4 destas são habilidades desenvolvidas ao se trabalhar o Pensamento Computacional:

- **Criatividade** é a habilidade de, tanto na vida profissional quanto na pessoal, inovar, ver novas soluções, criar novos recursos e novas invenções, e gerar novos serviços e negócios.
- **Pensamento crítico** é a habilidade de abordar os problemas de forma consistente e sistemática, compreender as ligações entre as ideias, determinar a importância e relevância dos argumentos e ideias, reconhecer, construir e avaliar argumentos, manter-se curioso sobre os diferentes assuntos e temas, refletir sobre a justificativa de suas próprias suposições, crenças e valores, e identificar inconsistências e erros de raciocínio, sendo assim capaz de mudar de ideia caso faça sentido.
- **Flexibilidade cognitiva** é a habilidade de resolver um problema de forma fácil e eficiente por meio do raciocínio fluido. Podemos dizer que é a capacidade de “pensar fora da caixa”.
- **Resolução de problemas complexos** é a habilidade de identificação de padrões, decomposição de problemas e análise do que é ou não relevante no estabelecimento dos passos para obter uma solução. Aqui é necessário ter maior nível de conhecimento em tecnologias, pois estas podem organizar o pensamento com mais eficiência e estratégia.

Nesse contexto, embora **programar** seja uma palavra que assuste a muitos, devemos sempre apresentar aos estudantes a possibilidade de **ensinar a máquina a fazer algo para nós**, ensinando-os a criar e a lidar com problemas e soluções que vão além do uso dos softwares de escritórios e suas aplicações, sendo que o mais importante é utilizar esse potencial para a solução de problemas do cotidiano e outros que ainda aguardam serem solucionados. Nesta perspectiva, Valente (2016) nos traz ainda que:

---

<sup>15</sup> WEF Fórum Econômico Mundial. Disponível em: <[http://gg.gg/weforum-org\\_top10\\_work-skills](http://gg.gg/weforum-org_top10_work-skills) >. Acesso em 12/2021.

“A ênfase nos conceitos da Ciência da Computação tem sido justificada com base no argumento que atividades realizadas no âmbito dessa ciência desenvolvem habilidades do pensamento crítico e computacional, e permitem entender como criar com as tecnologias digitais, e não simplesmente utilizá-las como máquinas de escritório.” VALENTE (2016, p.4).

Corroborando também com essa linha de raciocínio o autor Blikstein (2008), o qual nos afirma que:

“[...] os alunos aprendem “ciência computacional” mais facilmente do que ciência tradicional, por uma série de fatores cognitivos, epistemológicos e motivacionais. Boa parte da ciência e da matemática que ensinamos na escola foi inventada porque não tínhamos computadores, e seu aprendizado é desnecessariamente difícil, afastando qualquer aluno mais criativo.” BLIKSTEIN (2008, p.1).

Aprender computação é algo que fascina nossos estudantes, talvez por serem nativos digitais ou por perceberem a relevância e a necessidade do desenvolvimento de habilidades relacionadas à esta área. Dessa forma, segundo Brackmann (2017):

“É fundamental que todos os indivíduos tenham conhecimentos básicos de computação, já que todos estão mergulhados num universo permeado pelas tecnologias de informação e comunicação. Esses conhecimentos são tão significativos para a vida na sociedade contemporânea quanto aqueles em áreas tais como matemática, filosofia e física.” BRACKMANN (2017, p.17).

Portanto, a escola da atualidade deve acompanhar as novas tendências e, é tarefa do profissional da educação incluir no planejamento das aulas a inserção de atividades que proponham o desenvolvimento da criatividade, do pensamento crítico, da flexibilidade cognitiva e da resolução de problemas, seja por meio de atividade Maker, do desenvolvimento de Educação STEAM ou mesmo de atividades esparsas para que os aprendizes possam ter pleno desenvolvimento a partir de uma aprendizagem significativa para a vida.

### **3. A (não) abordagem das tecnologias digitais na BNCC e no Currículo do Espírito Santo e o letramento digital**

Esse capítulo pretende, discorrer acerca da BNCC e do Currículo das escolas, fazendo apontamentos para as necessidades de atualização destes documentos, além de mostrar algumas especificidades no que diz respeito ao Pensamento Computacional numa perspectiva de apresentação de apontamentos sobre a necessidade do letramento digital e a apropriação do tema para este letramento e o desenvolvimento de habilidades relacionadas à competência de valorizar e utilizar os conhecimentos historicamente construídos sobre o mundo físico, social, cultural e digital para entender e explicar a realidade. Nesta perspectiva, compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

#### **3.1 A BNCC e os necessários apontamentos para o desenvolvimento das habilidades do PC**

A Base Nacional Comum Curricular (BNCC) é um documento que define os conhecimentos, competências e habilidades que todos os estudantes devem desenvolver ao longo da Educação Básica (Educação Infantil, Ensino Fundamental e Ensino Médio), servindo apenas como um documento normativo, o qual deve ser complementado conforme a necessidade e suas especificidades, concretizando os direitos de aprendizagem prevista em legislação.

Dentro da perspectiva que nos interessa, a BNCC tem um foco muito grande no letramento matemático (o letramento matemático, segundo a BNCC, é definido pelas habilidades e competências para representar, raciocinar, comunicar e argumentar criticamente, com base nos conhecimentos da matemática) e ainda propõe que o estudante seja protagonista de seu conhecimento, a partir de uma matriz de competências que este deve consolidar para poder avançar com sucesso.

Nessa direção, Perrenoud (2000, p. 3) afirma que, “para desenvolver competências é preciso, antes de tudo, trabalhar por **problemas e projetos**, propor tarefas complexas e desafios que incitem os alunos a mobilizar seus conhecimentos e, em certa medida, completá-los.” Macedo (2005, p. 71), corroborando esse pensamento, escreve que “a competência é uma habilidade de ordem geral, enquanto a habilidade é uma competência de ordem particular, específica”.

As Habilidades são qualidades que o profissional tem para realizar alguma atividade. São aquelas características que podem ajudar um profissional a desenvolver competências. Assim, podemos dizer que a habilidade é inata. Sendo assim, fica evidente que é preciso implementar uma pedagogia ativa, aberta, cooperativa e com liberdade ao protagonismo.

Em uma perspectiva relacional, precisa-se tanto da habilidade como da competência e é essa que coordena, estrutura e articula os múltiplos fatores e requer que o professor consiga lidar, num determinado espaço de tempo, promovendo as aprendizagens.

Deste modo, dar aulas já não é mais o cerne da profissão do professor, portanto, ensinar, hoje, consiste em conceber, encaixar e regular situações de aprendizagens seguindo os princípios pedagógicos ativos e construtivistas. Assim, para os professores adeptos de uma visão construtivista e interacionista de aprendizagem, trabalhar no desenvolvimento de competências não é uma ruptura, uma vez que estas mobilizam conhecimentos, valores, atitudes e habilidades para a melhoria das relações do sujeito consigo mesmo e com os outros, para o alcance dos seus objetivos e para a tomada de decisões.

A BNCC apresenta dez competências gerais que precisam ser desenvolvidas visando ao desenvolvimento integral do sujeito, para que este se engaje na sociedade de forma mais participativa, construindo e comunicando conhecimentos. Essas competências referem-se ao desenvolvimento social, afetivo, cognitivo e emocional dos estudantes, e devem ser desenvolvidas de maneira intencional, incluindo-as nos planejamentos, pois essas podem tanto potencializar quanto atrasar as aprendizagens. Segundo Nóvoa (2019):

“A mudança faz-se a partir de uma matriz cultural e científica, afirmando a importância do conhecimento, sem ceder nem à ideologia do back to basics (a escola mínima de antigamente, do ler, escrever e contar), nem a uma escola folclórica afogada numa infinidade de projetos que, tantas vezes, apenas revelam a dificuldade para renovar as práticas pedagógicas. Não vale a pena alimentarmos ilusões, trazendo tudo para dentro da escola, uma escola transbordante, sem rumo e sem sentido. Mas vale a pena trabalhar para a construção de um espaço público de educação, a cidade educadora, no qual a escola se articula com outras instituições, grupos e associações.” (NÓVOA, 2019. p. 4).

Para melhor compreensão, destacamos a seguir as 10 Competências Gerais da Educação Básica, Figura 9, com atenção **às competências 2 e 5, que fomentam a necessidade de se utilizar e criar tecnologias:**

1. Valorizar e utilizar os conhecimentos historicamente construídos sobre o mundo físico, social, cultural e digital para entender e explicar a realidade, continuar aprendendo e colaborar para a construção de uma sociedade justa, democrática e inclusiva.
- 2. Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.**
3. Valorizar e fruir as diversas manifestações artísticas e culturais, das locais às mundiais, e também participar de práticas diversificadas da produção artístico-cultural.
4. Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.



5. **Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.**
6. Valorizar a diversidade de saberes e vivências culturais e apropriar-se de conhecimentos e experiências que lhe possibilitem entender as relações próprias do mundo do trabalho e fazer escolhas alinhadas ao exercício da cidadania e ao seu projeto de vida, com liberdade, autonomia, consciência crítica e responsabilidade.
7. Argumentar com base em fatos, dados e informações confiáveis, para formular, negociar e defender ideias, pontos de vista e decisões comuns que respeitem e promovam os direitos humanos, a consciência socioambiental e o consumo responsável em âmbito local, regional e global, com posicionamento ético em relação ao cuidado de si, dos outros e do planeta.
8. Conhecer-se, apreciar-se e cuidar de sua saúde física e emocional, compreendendo-se na diversidade humana e reconhecendo suas emoções e as dos outros, com autocrítica e capacidade para lidar com elas.
9. Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.
10. Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.

Dentre as competências expostas, a Competência 5 é bastante contundente em deixar aberta às possibilidades de uso e aplicação das ferramentas tecnológicas para que o professor consiga oferecer aos estudantes possibilidades de protagonismo, para que estes autonomamente desenvolvam conceitos, criem situações problemas e as solucionem.

Veja a seguir, o esquema que pode facilitar o entendimento dessas competências.

**Figura 9 - BNCC e as competências gerais**



Fonte: Plataforma Porvir<sup>16</sup>

Em relação à matemática, em específico, para as operações mentais e desafios de lógica, os estudantes devem transcender o entendimento e a compreensão dos conceitos, produzindo e elaborando problemas, ao invés de apenas resolvê-lo. Já com relação à tecnologia, esta deve se configurar como parte da aula e não apenas como metodologia ou ferramenta de auxílio docente, haja vista o surgimento crescente de novas tecnologias que nos possibilitam trabalhar problemas e projetos de forma diferenciada e atrativa.

Desta forma, o estudante deve hoje aprender sobre a tecnologia, os seus conceitos e modos de funcionamento, além de desenvolver competências e habilidades para poderem não somente saber manusear os aplicativos e equipamentos, mas sim saber como funcionam e conhecer os detalhes por trás de toda engenharia da ferramenta ou equipamento.

<sup>16</sup> Porvir é uma plataforma de conteúdos e mobilização sobre inovações educacionais, disponível em: <[http://porvir-prod.s3.amazonaws.com/wp-content/uploads/2018/02/28185234/BNCC\\_Competencias\\_Progressao.pdf](http://porvir-prod.s3.amazonaws.com/wp-content/uploads/2018/02/28185234/BNCC_Competencias_Progressao.pdf)>. Acesso em 03/2022.

O desenvolvimento das habilidades do pensamento computacional é fundamental desde as séries iniciais da educação básica, para que os nativos digitais possam apropriar-se de forma correta da compreensão, do uso e até criar tecnologias e soluções. Nesse contexto, segundo a BNCC:

“A área de Matemática, no Ensino Fundamental, centra-se na compreensão de conceitos e procedimentos em seus diferentes campos e no desenvolvimento do pensamento computacional, visando à resolução e formulação de problemas em contextos diversos. No Ensino Médio, na área de Matemática e suas Tecnologias, os estudantes devem consolidar os conhecimentos desenvolvidos na etapa anterior e agregar novos, ampliando o leque de recursos para resolver problemas mais complexos, que exijam maior reflexão e abstração.” (BNCC, 2017, p. 471).

Nesse sentido, fica implícito que deve-se incluir na grade curricular, desde o início da vida escolar do estudante, atividades que possibilitem o desenvolvimento de habilidades e competências tecnológicas, para posterior consolidação desses conhecimentos essenciais, haja vista que estes auxiliam na resolução, identificação, modelagem, análise e formulação de problemas, explorando de forma efetiva o raciocínio lógico, o espírito investigativo e a criatividade do aprendiz. Assim, estas habilidades foram esquematizadas, Figura 10, com intuito de facilitar a compreensão.

**Para conhecer mais detalhes da BNCC, acesse o site:**

<http://basenacionalcomum.mec.gov.br/historico>

<http://basenacionalcomum.mec.gov.br/abase/>

Figura 10 - Habilidades educacionais para o século XXI



Fonte: Daniel Sasaki, as seis habilidades para o século XXI<sup>17</sup>

### 3.2 O Currículo do ES e os necessários apontamentos para o desenvolvimento das habilidades do PC

A educação não pode ser reduzida ao ensino, portanto, é fundamental não vincular de forma engessada Currículo e Ensino. Daí, faz-se necessário redefinir e instituir o currículo como espaço de sentidos, evidenciando a cultura, o meio social, o mundo do trabalho e suas tecnologias.

<sup>17</sup> Daniel Sasaki, as seis habilidades para o século XXI, disponível em: <<https://www.researchgate.net/profile/Daniel-Sasaki/publication/344466546/figure/fig1/AS:942938737627137@1601825522481/Figura-1-As-seis-habilidades-para-o-seculo-XXI-que-todos-os-estudantes-devem-trabalhar.ppm>>. Acesso em 03/2022.

O Currículo do ES<sup>18</sup>, proposto como um documento que se alinha à BNCC, foi construído por professores da Educação Básica, além de incluir diversos outros profissionais na sua redação e estar em regime de colaboração com as diversas esferas da comunidade escolar. Esse documento foi um passo importante dado na construção do conhecimento e do processo de ensino e aprendizagem com a proposta de um documento curricular que reflete a diversidade do Estado do Espírito Santo. Contudo, o Currículo do ES é um documento que carece ainda de melhorias no direcionamento para o ensino e uso das ferramentas tecnológicas, como está escrito no próprio documento, e deve apresentar futuramente propostas com implementos que possibilitem essas melhorias pautadas no desenvolvimento individual dos estudantes e no contexto social em que vivem, levando em consideração os diversos fatores, dentre eles o consumo e uso das tecnologias existentes, que não existem.

Dentre essas melhorias necessárias ao Currículo do ES, cabe a nós, professores, a responsabilidade de contemplar a competência 5 da BNCC, qual seja, “compreender, usar e criar tecnologias de informação”, e de dar a oportunidade aos estudantes de desenvolverem habilidades referentes ao uso das ferramentas tecnológicas, oferecendo as condições necessárias para seu desenvolvimento, mesmo que, infelizmente, na maioria ou em todos os componentes curriculares, não tenham sido dadas a importância e a atenção necessárias a este tema.

Neste sentido, é preciso fortalecer e engajar os estudantes a desenvolverem habilidades e competências relacionadas às tecnologias, para poderem pleitear espaço na concorrência do mercado, e para isso é preciso que as atividades e os conteúdos curriculares propiciem o desenvolvimento do pensamento crítico, do protagonismo e da capacidade de resolução e criação inovadora de problemas e situações. Considerando o uso das tecnologias e as possibilidades que as mesmas proporcionam e ao mesmo tempo aproveitar seu potencial, levando em consideração o conhecimento adquirido, a realidade do contexto na qual a escola está inserida entre outras fragilidades e/ou potencialidades. Segundo Peters (2015):

---

<sup>18</sup> Currículo do ES é um documento, elaborado em regime de colaboração, disponível em: <<https://curriculo.sedu.es.gov.br/curriculo/#gsc.tab=0>>. Acesso em 06/2023. Serve como referencial e deve nortear as ações de ensino-aprendizagem nas escolas do Estado do ES.

“[...] é importante que se analise o nível de habilidade e competência dos estudantes como forma de identificar e retratar nós críticos que possam configurar como dificuldades na aprendizagem do estudante de modo a promover sua recuperação, oferecendo-lhes subsídio para avançar em seus estudos.” (PETERS, 2015, p. 8).

Além disso, as inovações tecnológicas difundidas na atualidade nas diversas áreas e setores faz com que se reiterem modelos novos e abrangentes de metodologias de ensino e aprendizagem dentro e fora de sala de aula. Nesse contexto, Fernandes (2015) nos ressalta que:

“[...] na prática escolar cotidiana esses modelos adquirem diversas caracterizações, podendo, até mesmo, muito frequentemente, coexistirem e se superpõem ao menos parcialmente. Boa parte dos professores costumam, inclusive, misturar princípios e métodos de dois ou mais modelos em suas práticas cotidianas, criando de certo modo modelos híbridos. Não há, pois, em boa parte das práticas escolares em ciências, ou até mesmo na maior parte delas, uma homogeneidade teórico-metodológica na realização de uma ou mais práticas sequenciais. Mais comum é o professor utilizar (ou misturar) diferentes modelos pedagógicos na sua lida diária profissional.” (FERNANDES, 2015, p. 34).

Portanto, é evidente a constatação da necessidade de implantação e prática de um currículo mais atualizado nas escolas, o qual deve se conectar às vivências e experiências externas a ele, ao mundo do trabalho e à cultura tecnológica da sociedade contemporânea. Como trazem Lopes e Macedo (2011), a perspectiva instrumental do conhecimento atua como propulsor dessa necessidade de mudança curricular, uma vez que um currículo baseado na perspectiva instrumental “(...) tem a finalidade de formar cidadãos capazes de gerar um benefício mais amplo para a sociedade” (LOPES E MACEDO, 2011, P.74). Sendo assim, o currículo não pode ser focado apenas em padronizar, em avaliar e em cumprir os programas de ensino e transmissão de conhecimento, logo é preciso tornar o estudante protagonista de seu próprio conhecimento e formar cidadãos capazes de gerar benefícios para a sociedade. Ainda nesta temática, Bacich e Moran acreditam que:

“A variedade de estratégias metodológicas a serem utilizadas no planejamento das aulas é um recurso importante, por estimular a reflexão sobre outras questões essenciais, como a relevância da utilização das metodologias ativas para favorecer o engajamento dos alunos e as possibilidades de integração dessas propostas ao currículo.” (BACICH E MORAN, 2018, p. 22).

Por fim, destacamos que conhecer o Currículo do ES e como se pretende que os estudantes desenvolvam as habilidades mínimas para se relacionar com o mundo do trabalho e com o meio social é tarefa do professor. Assim como, trabalhar o ensino e aprendizagem de conceitos e teorias da disciplina, correlacionando-os com as teorias da computação, seja por meio do uso do computador ou por meio de atividades desplugadas, também é uma responsabilidade do professor, visto que, embora o currículo básico não aponte a forma de se trabalhar o desenvolvimento de uma ou outra habilidade, é dever do profissional criar meios e formas de utilização de fundamentos da computação para solucionar problemas do dia a dia de forma prática, reflexiva, fácil e/ou lúdica, possibilitando assim que seus estudantes desenvolvam seus próprios meios e soluções, seja com uso de algoritmos, fluxogramas ou outros meios que facilitem e agilizem a forma de solucionar problemas de modo prático e ágil.

**Para conhecer todos os detalhes do Currículo do ES, acesse o site:**

<https://curriculo.sedu.es.gov.br/curriculo/documentos/>

### **3.3 Letramento e/ou educação digital**

O letramento e/ou educação digital corresponde ao uso de ferramentas e recursos tecnológicos em atividades didáticas e pedagógicas. E planejar o uso desses recursos com atividades que retém a atenção do público jovem em um mundo cada vez mais conectado é cada vez mais desafiador para o educador. Contudo, é uma oportunidade para a quebra de paradigmas na criação de oportunidades em usar e criar tecnologias dentro e fora de sala de aula, para que os estudantes alcancem um mínimo de **fluência digital**, a qual, segundo a Sociedade Brasileira de Computação, é a:

“[...] habilidade de encontrar, avaliar, produzir e comunicar informação usando plataformas digitais (englobando tanto diferentes dispositivos de hardware quanto de software). Se refere, por exemplo, ao uso de computadores e aplicativos, como software para formatar textos, produzir apresentações, buscar informações e insumos na internet, etc.” (SBC, 2018, p. 2).

Nesse sentido, fica bastante claro que uma das demandas do mundo contemporâneo é o letramento e a fluência digital, portanto, é necessário preparar os estudantes para saberem empregar essas ferramentas e recursos tecnológicos para além do uso, na criação e também como forma de trabalhar autonomia, segurança e uso consciente, apropriado, ético, reflexivo e criativo.

Vale ressaltar também que as vantagens em se adotar aulas digitais são muitas e uma delas é a flexibilidade, a qual permite ao estudante assistir à aula quando e onde quiser, sem a necessidade de realizar deslocamentos, além da praticidade, que permite a promoção da educação continuada e da capacitação profissional mesmo a distância. Deste modo, o alinhamento entre os conteúdos, as ferramentas didáticas e as ferramentas tecnológicas utilizadas e/ou que podem ser utilizadas no ensino dos conteúdos das disciplinas, e em especial de matemática, tem se mostrado cada vez mais necessário dentro e fora de sala de aula, visto que há diversos fatores que contribuem para que os estudantes sejam cativados e se mostrem mais interessados. Segundo Angela Bustos Kleiman:

“O letramento é complexo e abrange mais do que uma habilidade ou uma competência do sujeito que lê. É um processo que envolve diversas capacidades e conhecimentos em relação à leitura de mundo, o qual se inicia quando a pessoa começa a interagir socialmente com as práticas de letramento e o meio em que vive.” (KLEIMAN, 1995, p.20).

Além disso, a formação de cidadãos plenos e conscientes na era digital depende do bom uso da internet e das ferramentas e recursos tecnológicos disponíveis, e é justamente através da tecnologia e da democratização do acesso à informação que se concretiza um letramento digital abrangente, eficiente e escalável. Na Figura 11, é possível ver como algumas competências estão relacionadas com o letramento digital.



Figura 11 - Letramento digital



Fonte: Aliança Brasileira pela Educação<sup>19</sup>

Contudo, ao tratarmos do letramento digital, é preciso reconhecer também que é notável que a escola, por meio educador, e para consolidação de aprendizagem significativas mediante o letramento digital, possui grande responsabilidade diante da falta de estrutura das salas de aulas, da limitação de recursos e da formação profissional defasada, entre outras limitações que comprometem a qualidade do trabalho pedagógico na contemporaneidade.

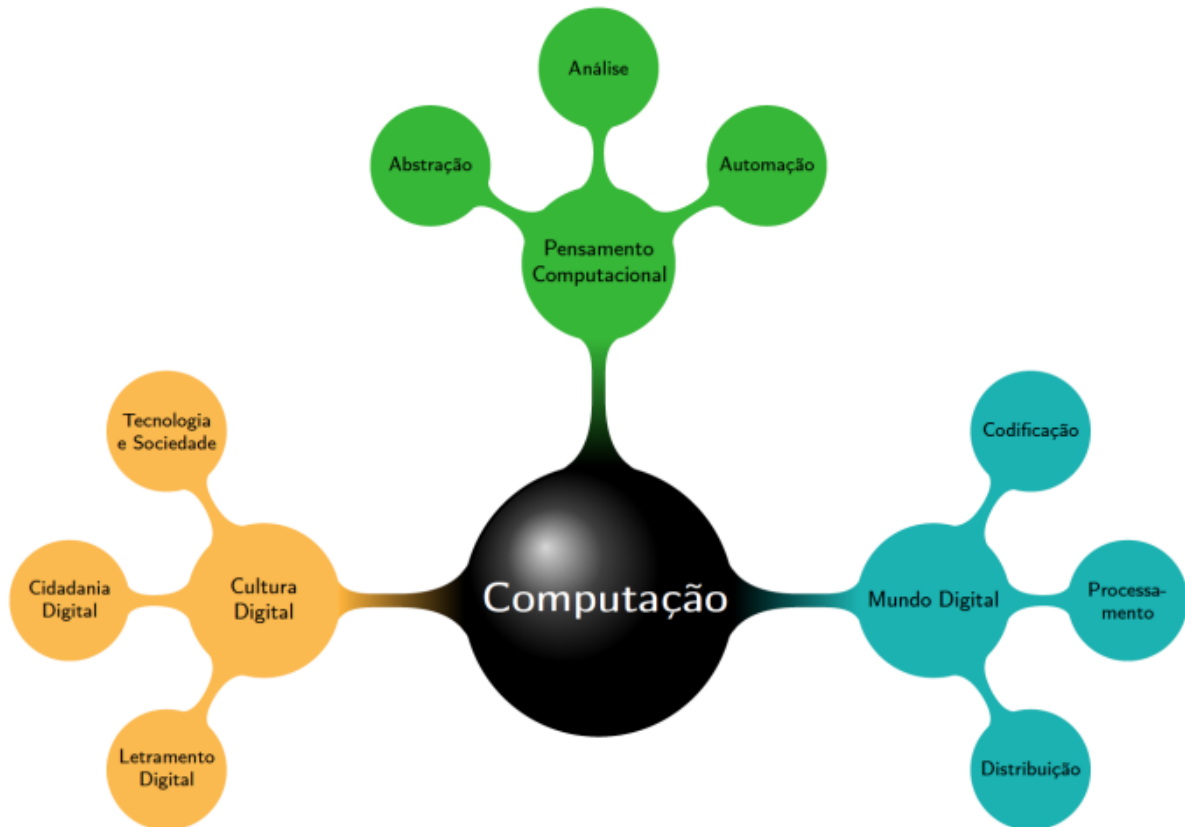
Para tanto, o Centro de Inovação para a Educação Brasileira (CIEB)<sup>20</sup> preparou um material que é considerado o Currículo de Referência em Tecnologia e Computação e que tem como principal objetivo apoiar redes de ensino e escolas a incluir os temas tecnologia e computação em suas propostas curriculares. O Currículo está organizado em três eixos: Cultura Digital, Pensamento Computacional e Tecnologia Digital, cada um com seus conceitos, Figura 12. Ao todo, são dez conceitos e 147 habilidades.

<sup>19</sup> Fonte: Aliança pela educação, disponível em: <[https://aliancapelaeducacao.com.br/index.php?option=com\\_content&view=article&id=616:a-importancia-da-educacao-digital-no-ensino-contemporaneo&catid=157&Itemid=849](https://aliancapelaeducacao.com.br/index.php?option=com_content&view=article&id=616:a-importancia-da-educacao-digital-no-ensino-contemporaneo&catid=157&Itemid=849)>. Acesso em 03/ 2022.

<sup>20</sup> CIEB - Centro de Inovação para Educação Brasileira. Disponível em: <<https://cieb.net.br/cieb-lanca-curriculo-de-referencia-em-tecnologia-e-computacao/>>. Acesso em 12/2021.

Nesse sentido, é preciso que a escola repense suas práticas, o ensino e a possibilidade de letrar seus estudantes sob a perspectiva da necessária urgência da sociedade em desenvolver habilidades que contemplem o uso e as potencialidades das Tecnologias Digitais de Informação e Comunicação (TDICs), com um letramento digital eficaz e abrangente.

Figura 12 - Eixo das diretrizes da SBC 2018



Fonte: SBC 2018<sup>21</sup>

**Para conhecer mais deste importante Currículo, acesse:**

<https://curriculo.cieb.net.br/>

<sup>21</sup> SBC 2018, Sociedade Brasileira de Computação, disponíveis em: <<https://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1220-bncc-em-itinerario-informativo-computacao-2>>. Acesso em 12/2022.

O desenvolvimento das habilidades do PC e do letramento digital é apresentado na BNCC para não ser contemplado em toda sua importância na vida e na sociedade contemporânea. Nesse sentido, falta o enaltecimento do desenvolvimento da computação e de suas possibilidades e isso tem sido criticado por muitos autores. A exemplo disso, um artigo publicado por Suellen Rodolfo Martinelli, Luciana Aparecida Martinez Zaina e Tiemi Christine Sakata, no VII Congresso Brasileiro de Informática na Educação (CBIE 2019) - “MultiTACT: uma abordagem para a construção de atividades de ensino multidisciplinares para estimular o Pensamento Computacional no Ensino Fundamental I” - nos mostra, como podemos ver na tabela abaixo, Figura 13, a relação das diversas áreas do conhecimento com as respectivas habilidades pelas quais há pertinência de se trabalhar o PC, sendo evidente que o seu trabalho multidisciplinar cria mais expectativas e abre possibilidades para as diversas áreas de conhecimento.

**Figura 13 - Códigos identificados na BNCC, conforme a área de conhecimento**

Área de Matemática	Área de Linguagens	Área de Ciências da Natureza	Área de Ciências Humanas
(MA01) Interpretação e manipulação de dados em tabelas.	(LP01) Reconhecer diferentes gêneros textuais.	(CI01) Identificar constelações.	(GE01) Aspectos sobre economia, processos de produção e consumo.
(MA02) Aplicação de medida padronizada.	(LP02) Reconhecer letras e/ou compor palavras.	(CI02) Reciclagem e identificação de recicláveis.	(GE02) Pontos cardiais e de referência.
(MA03) Problemas sobre situações comerciais e sistema monetário.	(LP03) Interpretação de linguagem verbal e não verbal.	(CI03) Plantas e etapas de plantio.	(GE03) Interpretação e/ou representação de espaços e mapas.
(MA04) Aplicabilidade das operações fundamentais.	(LP04) Produção e/ou interpretação colaborativa de texto.	(CI04) Identificar e comparar propriedades entre materiais/objetos.	(HI01) Papéis sociais das pessoas nas comunidades.
(MA05) Identificar e representar frações.	(AR01) Explorar brincadeiras regionais.	(CI05) Construção de dispositivos óticos.	(HI02) Comparação de diferentes linguagens e seus aspectos culturais.
(MA06) Estratégias de cálculo mental e estimativas.	(AR02) Uso de softwares e multimídias na produção/criação artística.		
(MA07) Quantificação, agrupamento e ordenação de objetos.	(AR03) Uso de diferentes técnicas e materiais na composição de artes visuais.		
(MA08) Trajetos e pontos de origem/destino.			
(MA09) Deslocamentos e mudanças de direção, sentido e giro.			
(MA10) Composição de formas geométricas e/ou análise de seus atributos.			
(MA11) Solução de um mesmo problema com diferentes algoritmos.			

Fonte: Adaptado de, Workshops do Congresso Brasileiro de Informática na Educação<sup>22</sup>.

**Para conhecer mais sobre as possibilidades do MultiTACT, acesse:**

<http://gg.gg/MultiTACT-atividades-PC>

ou

<https://repositorio.ufscar.br/handle/ufscar/11199>

<sup>22</sup> Anais dos Workshops do Congresso Brasileiro de Informática na Educação, disponível em: <<http://gg.gg/MultiTACT-atividades-PC>>. Acesso em 03/2022.

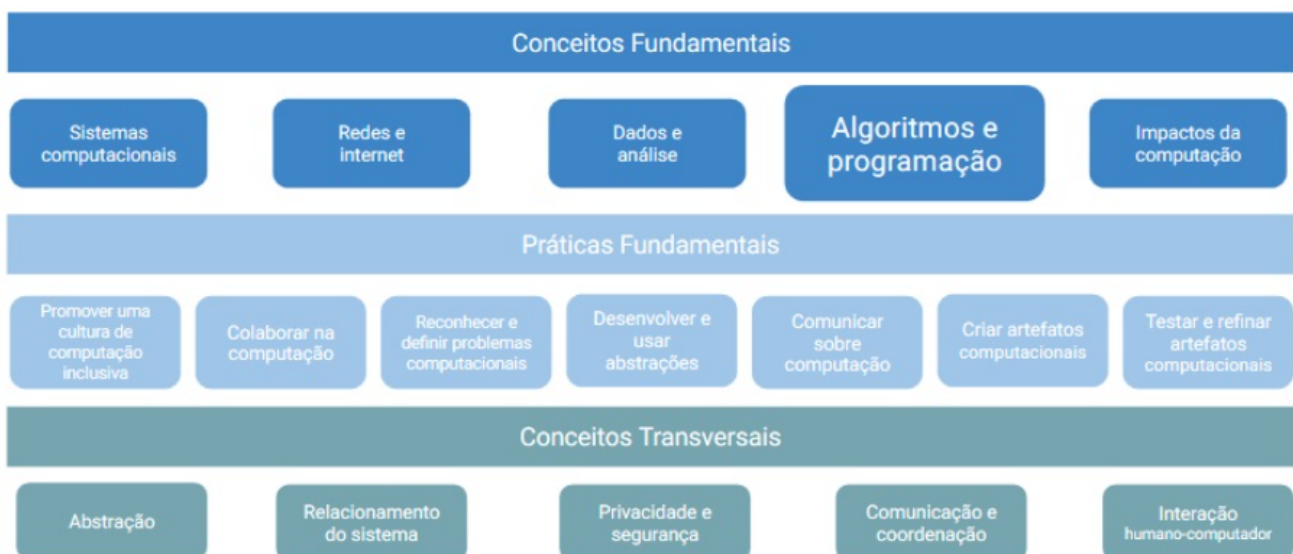
Desta forma, considerando a importância de entendermos a computação e a necessidade da inserção desta no ambiente escolar para o desenvolvimento adequado das habilidades do PC, apresentaremos aqui, na Figura 14, alguns conceitos fundamentais para o entendimento desta área e que se tornaram comuns no mundo contemporâneo, aos nativos digitais. A sistematização está apresentada em três camadas:

**Conceitos fundamentais** - Sistemas computacionais; Redes e internet; Dados e análise; Algoritmos e programação; Impactos da computação.

**Práticas fundamentais** - Promover uma cultura de computação inclusiva; colaborar na computação; reconhecer e definir problemas computacionais; desenvolver e usar abstrações; comunicar sobre computação; criar artefatos computacionais; testar e refinar artefatos computacionais.

**Conceitos transversais** - Abstração; Relacionamento do sistema; privacidade e segurança; comunicação e coordenação; interpretação humano-computador.

Figura 14 - Conceitos fundamentais das tecnologias digitais



Fonte: Galeria UFSC<sup>23</sup>

<sup>23</sup> Galeria UFSC, disponível em: <<https://galeria.ufsc.br/d/193702-1/imagemCSTA.jpg>>. Acesso em 12/ 2021.

## 4. Fluxogramas, algoritmos e linguagens de programação

Neste capítulo, iremos apresentar os conceitos de fluxograma e algoritmo trazidos pela BNCC e a SBC, além de apresentar alguns dos tipos de linguagens de programação existentes mais utilizados e, abordar quais deles seriam mais adequados para o desenvolvimento de um determinado projeto.

Este capítulo traz, também, os conceitos importantes para o desenvolvimento das atividades que compõem as propostas didáticas, que serão apresentadas nas oficinas didáticas dentro dos circuitos de aprendizagem no capítulo 6.

### 4.1 Fluxogramas

Os fluxogramas começaram a ser utilizados em 1921, quando Frank Gilberth apresentou, pela primeira vez, um desenho de processo a fim de encontrar o melhor caminho para aumentar a produtividade na indústria. Atualmente, segundo o dicionário Michaelis, fluxograma é a “representação gráfica, por meio de símbolos geométricos próprios, que apresenta o fluxo ou a sequência normal de um procedimento, problema ou sistema, que permite visualizar, por meio de ilustrações, de maneira sintética, as diversas etapas de um processo” (FLUXOGRAMA, 2022).

Quanto à área de informática, o conceito de fluxograma é de um diagrama para representação de um algoritmo, ou seja, um diagrama de fluxo representado por um conjunto de símbolos com significados e representações bem específicas, uma linguagem visual. Já a BNCC salienta que:

“Associado ao pensamento computacional, cumpre salientar a importância dos algoritmos e de seus fluxogramas, que podem ser objetos de estudo nas aulas de Matemática. Um algoritmo é uma sequência finita de procedimentos que permite resolver um determinado problema. Assim, o algoritmo é a decomposição de um procedimento complexo em suas partes mais simples, relacionando-as e ordenando-as, e pode ser representado graficamente por um fluxograma.” (BNCC, 2017, p. 271).

Embora a BNCC aponte que os fluxogramas podem ser objetos de estudo nas aulas de matemática, em específico, vários outros estudos já têm nos mostrado que estes auxiliam nas mais diversas áreas de conhecimento, com aplicações diretas e indiretas nas soluções diversas de problemas. Além disso, atualmente, há vários sites e aplicativos gratuitos que auxiliam na criação dos fluxogramas, assim como também na escrita de algoritmos. Neste sentido, a SBC nos traz que:

“Existe uma gama de outras linguagens visuais que podem ser empregadas para este fim, e que possuem características muito mais desejáveis do ponto de vista didático-pedagógico.” (SBC, 2018, p. 3).

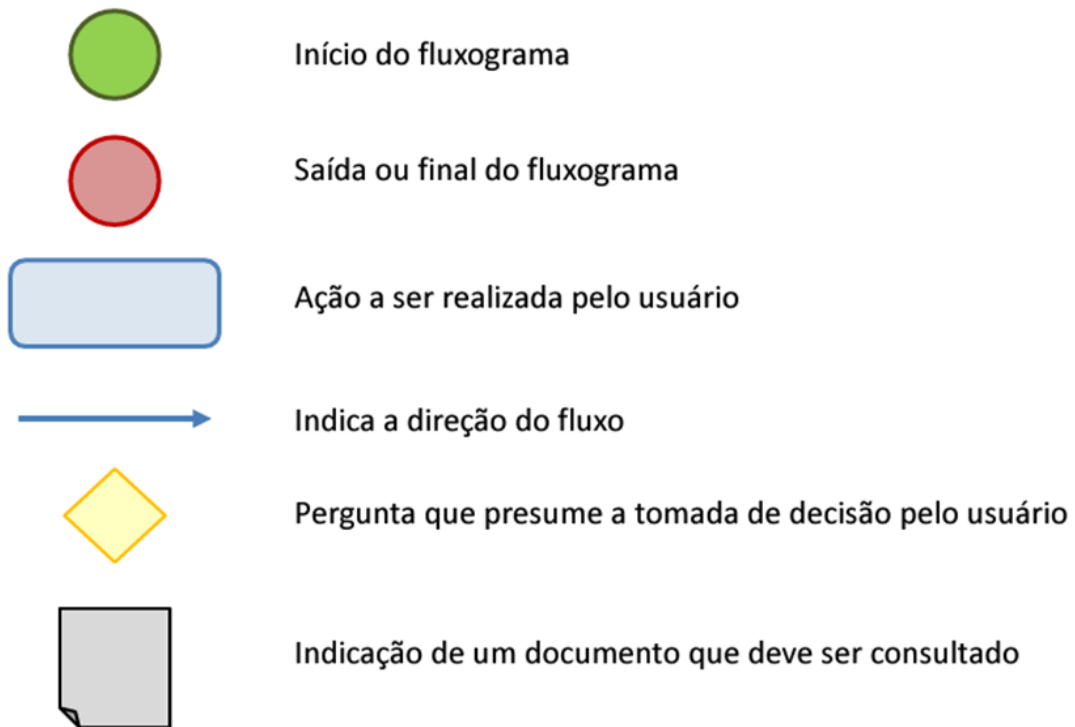
Entretanto, é importante que atentemos aos nossos objetivos quanto ao uso de fluxogramas, visto que o desenvolvimento das habilidades do pensamento computacional por meio da resolução de problemas pode abrir um leque muito amplo quando se trata de fluxogramas, o que pode levar até a questões mais técnicas. No entanto, isso não significa que pessoas não especialistas não consigam trabalhar esta temática. Sendo assim, o que se questiona é qual o melhor tipo de linguagem a se utilizar, qual o conceito de fluxograma abordado e como se dará o seu uso em sala de aula, assim como a forma que iremos trabalhar e direcionar a abordagem do ensino e aprendizagem de fluxogramas e algoritmos, uma vez que também não devemos confundir ambos, já que há diferenças entre seus conceitos e usos nas diversas abordagens para a resolução de problemas.

Isso remete, mais uma vez, aos objetivos do desenvolvimento do Pensamento Computacional e como os algoritmos e também os fluxogramas são importantes nesse processo de ensino e aprendizagem e no desenvolvimento das habilidades em solucionar problemas com aplicação de linguagens visuais, Figura 16, que facilitam o entendimento e compreensão de processos complexos.

Portanto, pode-se concluir que há várias opções de linguagens visuais e também de ferramentas que podem auxiliar a profissionais e aprendizes no desenvolvimento de projetos e soluções com o uso de fluxogramas e algoritmos; por isso, é fundamental sabermos quais dessas ferramentas (e/ou quais dessas linguagens) são mais adequadas a cada projeto ou solução.

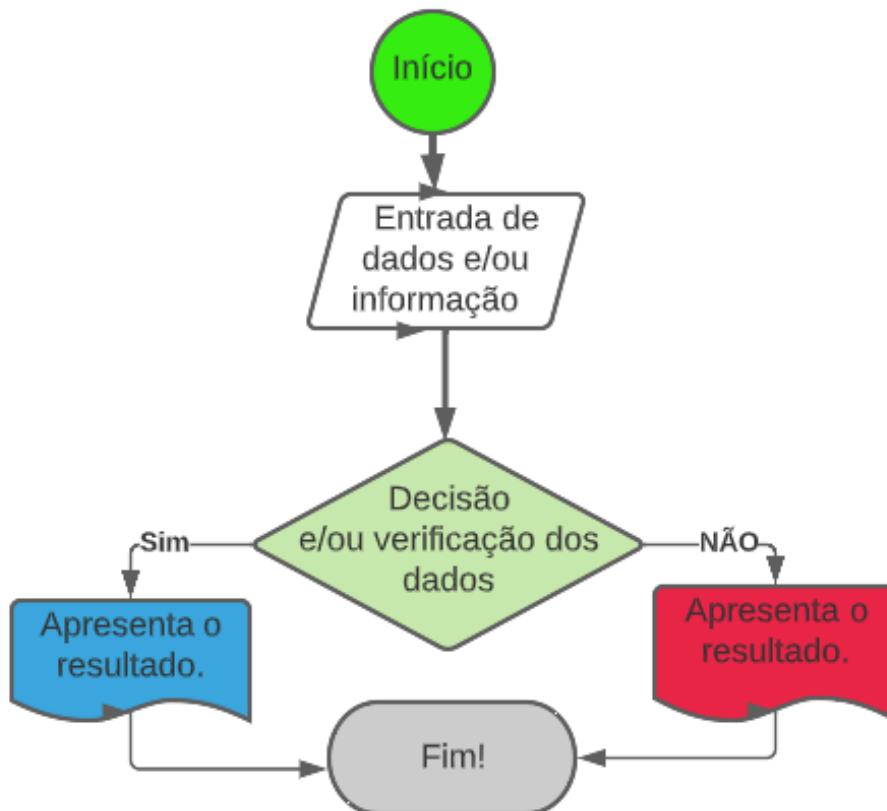
Por fim, para compreendermos melhor os fluxogramas, temos alguns dos símbolos mais utilizados em sua escrita e um exemplo deste, Figura 15:

Figura 15 - Principais símbolos dos fluxogramas



Fonte: Autoria própria, 2022.

Figura 16 - Entrada de dados para fluxogramas



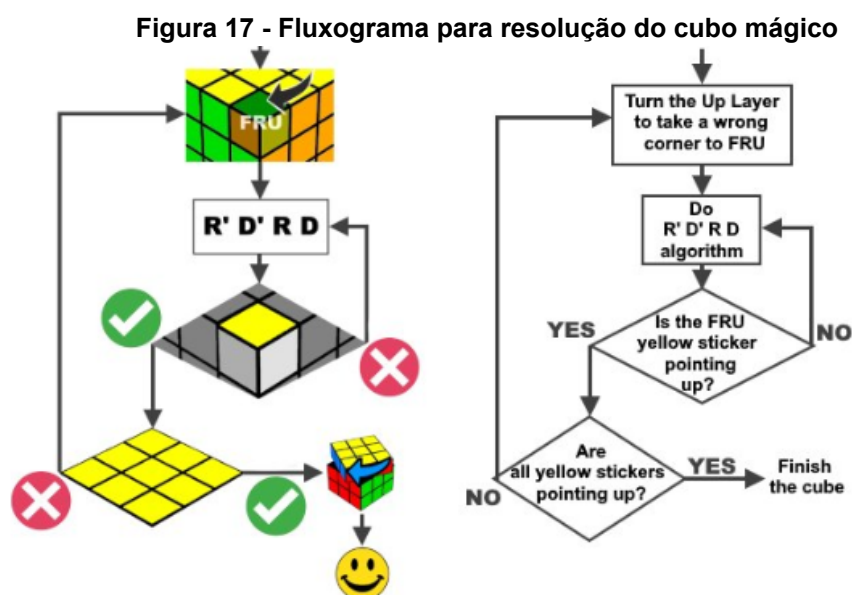
Fonte: Autoria própria, 2022.

## 4.2 Utilização e aplicações de fluxogramas

Os fluxogramas aparecem nas habilidades de matemática dentro do Currículo do Espírito Santo como forma de trabalhar e desenvolver as habilidades do pensamento computacional. Dentre as inúmeras aplicações, os fluxogramas auxiliam em processos de organização e memorização de conteúdos e ajudam na solução de problemas. Por isso, é uma ferramenta importante para o ensino e aprendizado. Como exemplo, listamos alguns propósitos para os quais são usados em diversas áreas:

- Definir um processo;
- Modelar um processo;
- Identificar desperdícios;
- Padronizar um processo;
- Identificar gargalos;
- Melhorar um processo.

Sendo assim, a Figura 17, tradução na Figura 18, traz um fluxograma que apresenta uma solução do **cubo mágico**, um quebra cabeça tridimensional, também conhecido como Cubo de Rubik (em inglês, Rubik's Cube), inventado pelo professor de arquitetura húngaro Ernő Rubik em 1974. Este exemplo nos mostra quanto um fluxograma auxilia na visualização e solução de problemas complexos.

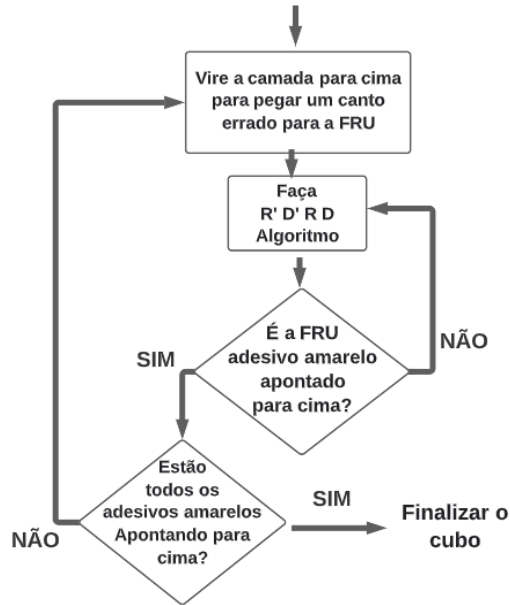


Fonte: Ruwix<sup>24</sup>.

<sup>24</sup> Ruwix, disponível em: <<https://ruwix.com/pics/solution/finish-solution-rubiks-cube.jpg>>. Acesso em 03/2022.



Figura 18 - Fluxograma para resolução do cubo mágico, tradução



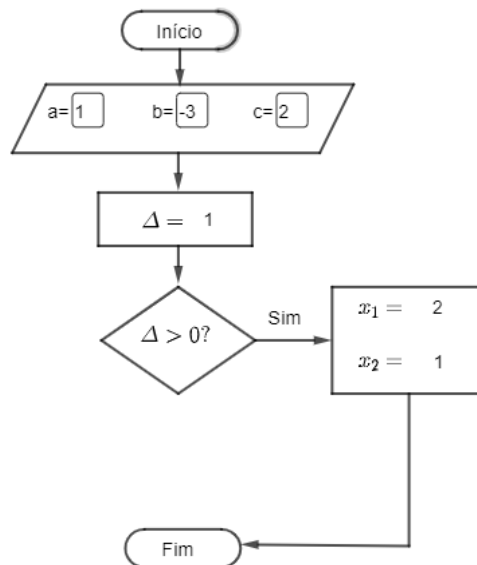
Legenda: O que as letras significam nos algoritmos:  
F: frente, R: direita, U: cima, L: esquerda, D: baixo

Fonte: Autoria própria, 2022.

Outros três exemplos de aplicação do fluxograma são:

- Fluxograma, Figura 19. Traz uma maneira de mostrar aos estudantes o fluxo que deve-se seguir para solucionar uma equação quadrática com a utilização da fórmula de Bháskara.

Figura 19 - Fluxograma para resolução de equação do 2º grau



Fonte: Autoria própria, 2022. Adaptado de Dênis Vargas, Geogebra.org<sup>25</sup>.

<sup>25</sup> Dênis Vargas, disponível em: <<https://www.geogebra.org/m/ccxpxydg>>. Acesso em 04/2022.

- b) Fluxograma para resolução de problemas de geometria, Figura 20. O fluxo que deve-se seguir para desenhar um quadrado.

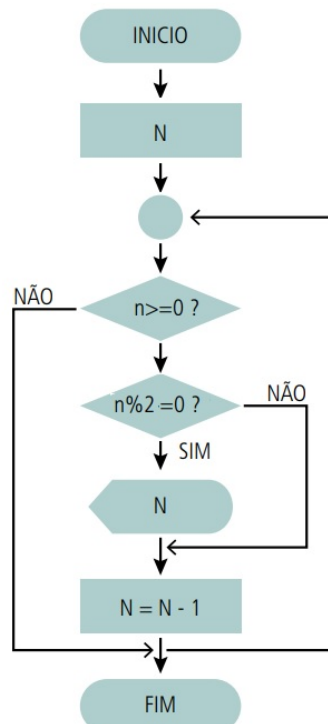
Figura 20 - Fluxograma construção de um quadrado



Fonte: Autoria própria, 2022. Aplicativo, adaptado de Marco A. Manetta<sup>26</sup>.

- c) Fluxograma que imprime todos os números pares positivos e menores do que N, Figura 21.

Figura 21 - Fluxograma para imprimir números pares positivos menores que N



Fonte: Adaptado de Sampaio 2008.

<sup>26</sup> Geogebra, disponível em: <<https://www.geogebra.org/m/ymjem8zs>>. Acesso em 05/2022.

### 4.3 Algoritmos

Segundo o dicionário Michaelis, algoritmo é um “processo de cálculo que, por meio de uma sequência finita de regras, raciocínios e operações, aplicada a um número finito de dados, leva à resolução de grupos análogos de problemas” (ALGORITMO, 2021). Já Guimarães (1994) traz a seguinte definição:

“Um algoritmo nada mais é que um texto contendo comandos (instruções) que devem ser executados numa determinada ordem. Esse texto em si não nos interessa, mas, sim, seu significado, ou seja, aquilo que ele representa.” (GUIMARÃES; LAGES, 1994, *Apud* BAUDSON, p. 15).

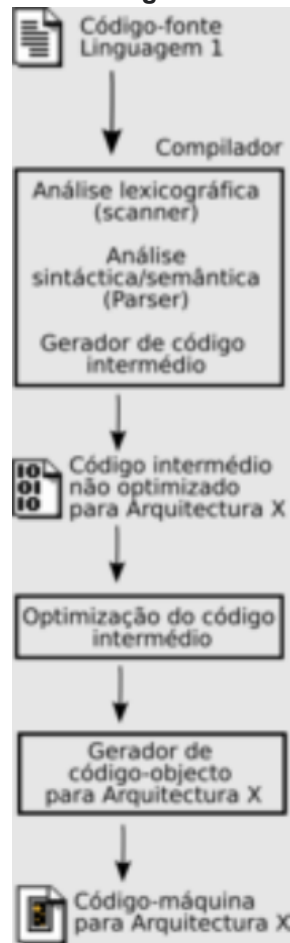
Deste modo, de uma forma mais simples, o algoritmo é a própria sequência de instruções para a execução de uma tarefa, isto é, é uma receita ou roteiro que indica todos os procedimentos necessários para realizar algo ou resolver algum problema.

Assim, definido já o conceito, ainda se faz necessário, no campo da programação e do desenvolvimento de um algoritmo, conhecer outros dois conceitos importantes para um desenvolvedor que pretende escrever um, quais sejam:

- **Compiladores:** são programas especializados em converter as instruções escritas na linguagem de programação em linguagem de máquina.
- **Ambiente de programação:** antes de iniciarmos a escrita ou edição, precisamos de um editor de texto e de um compilador instalado, o que configura o ambiente de programação.

Vale ressaltar aqui que, na atualidade, com o avanço tecnológico, tornou-se comum o uso de IDEs (Ambiente de Desenvolvimento Integrado), que são ambientes integrados que possuem editor de texto e também ajudam na correção da sintaxe, na utilização de bibliotecas e na compilação dos programas, tudo em um único lugar. O fluxograma, Figura 22, mostra esse fluxo de funcionamento:

Figura 22 - Fluxograma de uma IDE



Fonte: Wikipedia.org<sup>27</sup>.

#### 4.4 Linguagens de programação e a sua evolução

Inicialmente, as linguagens de programação eram códigos matemáticos, os quais, com a evolução da lógica matemática e com a abstração de conceitos mais complexos, tornaram a resolução de problemas complexos mais simples e prática, o que tem facilitado o mundo do trabalho e as relações sociais de modo geral. Essas linguagens possuem várias tipologias, mas todas seguem um mesmo padrão e/ou um algoritmo, que pode ser definido como uma sequência finita de passos (instruções) para resolver um determinado problema. Assim, ao desenvolvermos um algoritmo, estamos estabelecendo um padrão de comportamento, ou uma norma de execução de ações, que deverá ser seguido, para alcançar o resultado esperado quanto a um problema.

<sup>27</sup> Disponível em: <<https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Nt-compilador.png/100px-Nt-compilador.png>>. Acesso em 22/11/2022.

Essas evoluções das linguagens de programação e da lógica matemática apresentam pontos de convergência, de modo que o desenvolvimento dos programas de computadores montados para resolverem problemas matemáticos usam a lógica matemática, assim, quando você aprende uma linguagem e/ou seus códigos e comandos, está aprendendo a utilizar também conceitos da matemática, da mesma forma que, quando você desenvolve as habilidades e adquire conhecimento sobre um determinado assunto, aquilo que era complexo já não parece tão difícil quanto era no primeiro contato. Desse modo, a partir do momento em que você se dedica, passa a conhecer e obtém ferramentas para usar de forma adequada, torna-se mais fácil lidar com os problemas e as soluções, e isso acontece na área de matemática, assim como em outras áreas e situações.

Quando se fala em programação e software, é muito comum imaginarmos isso como algo difícil e/ou muito distante do dia a dia escolar. Mas, não é bem assim, afinal, hoje já existem linguagens de programação bem simples e práticas, por meio das quais é possível programar mesmo sem conhecer muito sobre programação. Você pode ver, por exemplo, que a programação de um carrinho de controle remoto se torna simples com a ferramenta adequada. No entanto, isso não significa que precisamos ser peritos em TI, mas estudar um pouco mais sobre essas diversas linguagens e ferramentas para nos apoderarmos dessas novas formas de solucionar problemas e tendências tecnológicas que podem nos auxiliar no dia a dia.

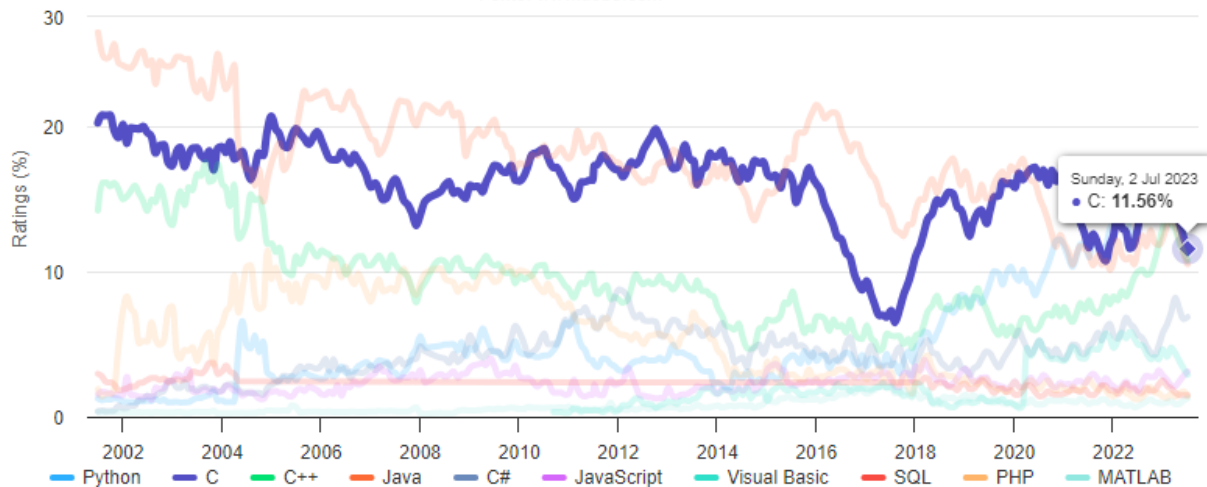
Na atualidade, com o desenvolvimento de plataformas e ferramentas e com a existência das muitas linguagens de programação que têm se perpetuado e se aperfeiçoado, é possível a uma pessoa leiga em conhecimentos de computação desenvolver um programa ou sistema. Há diversas linguagens de programação e, no site TIOBE é possível acompanhar quais são as que estão sendo mais usadas no mundo. Assim, o índice da comunidade de programação TIOBE<sup>28</sup>, Figura 23, é um indicador da popularidade das linguagens de programação, com atualizações mensais.

---

<sup>28</sup> Tiobe, disponível em: <<https://www.tiobe.com/tiobe-index/>>. Acesso em 07/2023.

**Figura 23 - Índice da comunidade de programação TIOBE**  
**Índice da Comunidade de Programação TIOBE**

Fonte: [www.tiobe.com](http://www.tiobe.com)



É importante observar, porém, que o índice TIOBE não trata de qual é a melhor linguagem de programação ou da linguagem na qual a maioria das linhas de código foi escrita. Antes, o índice pode ser usado para verificar se suas habilidades de programação continuam atualizadas ou para tomar uma decisão estratégica sobre qual linguagem de programação deve ser adotada ao iniciar a construção de um novo sistema de software.

No índice TIOBE, a linguagem C aparece entre as primeiras posições e seu uso, dentre as linguagens para sistemas embarcados (sistemas embutidos em microprocessadores, como a placa Arduino, por exemplo), é predominante e tem seu espaço garantido por muito tempo, dadas as características e requisitos exigidos pela maioria dos projetos eletrônicos atuais. Além disso, ela tem sido utilizada também para criar novas linguagens.

Por fim, vale destacar aqui que o primeiro passo para se aprender a programar é conhecer sobre a lógica da programação, isto é, sobre a técnica utilizada para desenvolver instruções em uma sequência organizada e coesa para atingir determinado objetivo e/ou a resolução de um problema, usando pseudocódigo, fluxograma e teste de mesa. Resumindo, é a organização e planejamento de instruções, em um algoritmo, visando tornar viável a implementação de um programa ou software.

## IMPORTANTE

Não se assuste com as diversas formas de programar. Na atualidade foram desenvolvidas linguagens que facilitam muito a programação, que são as linguagens em blocos lógicos. Por meio dessas linguagens, é possível ensinar as crianças desde pequenas a criarem seus códigos de forma fácil e rápida, sem a necessidade de ser um especialista do TI.

Para quem quer aprender mais e se aprofundar, conheça 15 plataformas que oferecem cursos gratuitos de programação:

- 1. Coursera
- 2. edX
- 3. Code Academy
- 4. BitDegree
- 5. Code.org
- 6. Khan Academy
- 7. Udemy
- 8. Codewars
- 9. Free Code Camp
- 10. MIT Open Courseware
- 11. W3 Schools
- 12. Bento.io
- 13. Upskill
- 14. Codeasy.net
- 15. Code Aveng

### 4.5 Tipos de linguagens de programação

Os diferentes tipos de linguagens de programação existentes, mostram como a necessidade em busca de facilidade na interação humano computador tem avançado e nesse sentido, apresentamos aqui algumas das linguagens que poderão ser úteis no desenvolvimento das oficinas didáticas propostas no capítulo 6 deste trabalho. O objetivo em apresentar alguns dos tipos de linguagens de programação é, aguçar a curiosidade e despertar o interesse do leitor a pesquisar um pouco mais sobre o assunto e assim, possa adequar seus planos e projetos conforme a necessidade.

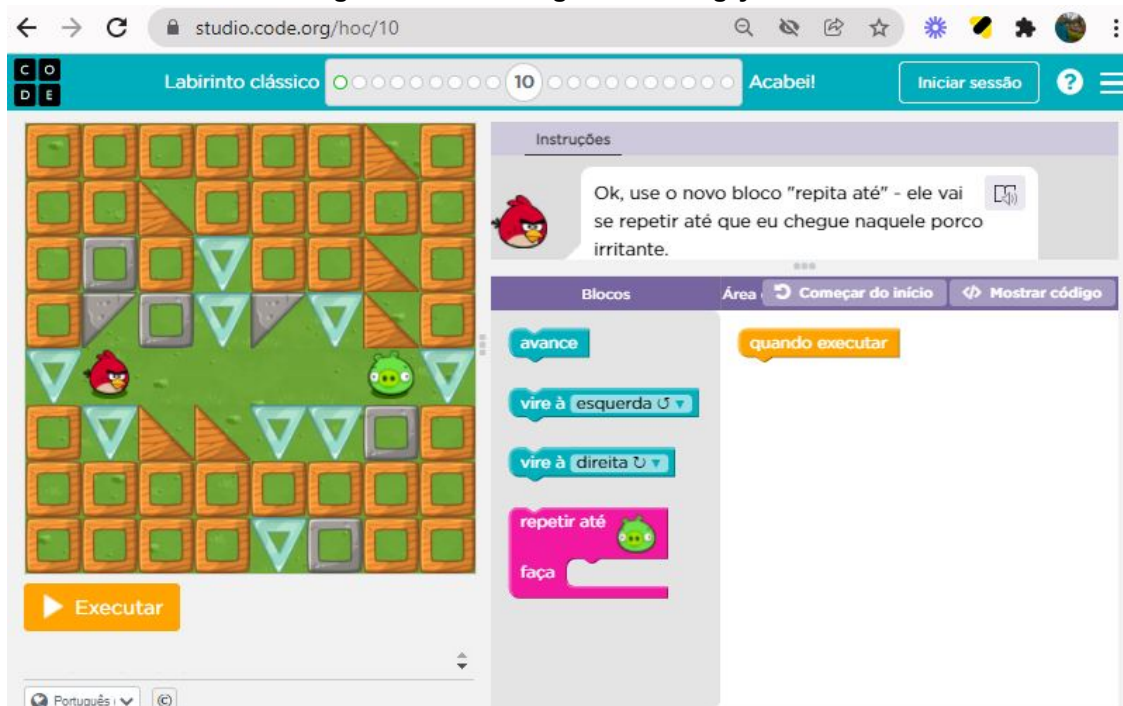
#### 4.5.1 Linguagem em bloco

Visando dar às pessoas a oportunidade de desenvolverem programas desde cedo, especialmente as crianças, assim como a capacidade de resolver problemas usando a tecnologia, a criatividade e a ludicidade, foram criados diversos programas que possibilitam a manipulação de uma linguagem de programação por meio de blocos de comandos, por meio dos quais o usuário do ambiente de programação define a sua estrutura lógica.

#### 4.5.2 Code.org (Angry Birds)

Para crianças e até mesmo adultos que não possuem conhecimentos em programação e software, é indicado iniciar os primeiros passos nesta área com a ajuda do Angry Bird, da Code.Org, o qual ajuda, de forma lúdica e interativa, no desenvolvimento da lógica de programação pela qual o usuário interage com o Angry Bird em um labirinto. Assim, o objetivo do primeiro programa é fazer com que um Angry Bird atravesse um labirinto para apanhar o porquinho, Figura 24.

Figura 24 - Code.org desafio Angry Birds



Fonte: Página Code.org<sup>29</sup>.

<sup>29</sup> Code.org, disponível em: <<https://studio.code.org/hoc/1>>. Acesso em 05/2022.



### 4.5.3 Scratch

O Scratch é um projeto do grupo Lifelong Kindergarten, idealizado por Mitchel Resnick e desenvolvido no Media Lab do Instituto de Tecnologia de Massachusetts. Ele é uma ferramenta de apoio à aprendizagem, isto é, um software que se utiliza de blocos lógicos e itens de som e imagem para você desenvolver suas próprias histórias interativas, jogos e animações, além de compartilhar de maneira online suas criações.

Segundo o próprio site do Scratch, Figura 25, ele foi projetado especialmente para usuários com idade entre 8 e 16 anos, mas é usado por pessoas de todas as idades e em mais de 150 países, estando disponível em mais de 40 idiomas e sendo fornecido gratuitamente para os principais sistemas operacionais (Windows, Linux e Mac).

A capacidade de escrever programas de computador é uma parte importante da alfabetização digital na sociedade de hoje. Deste modo, quando as pessoas aprendem a programar usando o Scratch, elas aprendem estratégias importantes para a resolução de problemas, projetos de design e comunicação de ideias.

Para conhecer mais sobre esta importante ferramenta, acesse o site ScratchBrasil<sup>30</sup>, conheça os tutoriais Scratch que auxilia os iniciantes e até dá dicas para quem já conhece a ferramenta.

**Figura 25 - Mascote Scratch**



Fonte: Página Scratch MIT Massachusetts<sup>31</sup>.

<sup>30</sup> Página ScratchBrasil, disponível em: <<http://www.scratchbrasil.net.br/index.php/materiais/tutoriais.html>>. Acesso em 05/2022.

<sup>31</sup> Página Scratch MIT Massachusetts, disponível em: <<https://scratch.mit.edu/>>. Acesso em 05/2022.

#### 4.5.4 Pictoblox

O Pictoblox, Figura 26, é um programa gratuito, com interface gráfica baseada em programação em blocos. Ele é ideal para os iniciantes darem os primeiros passos no mundo da programação, pois sua interface amigável tem a funcionalidade de ‘arrastar e soltar’ que elimina a necessidade de memorizar sintaxe e regras para o desenvolvimento da programação de jogos, animações, histórias e músicas. Além disso, pretende focar na solução de problemas e no desenvolvimento das habilidades do Pensamento Computacional, e em seu ambiente é possível encontrar vários projetos que podem ser usados como modelos para construção de projetos similares com objetivos em comum.

Figura 26 - PictoBlox



Fonte: Página PictoBlox<sup>32</sup>.

<sup>32</sup> PictoBox é uma ferramenta baseada no Scratch que permite o desenvolvimento da programação e conexão com microcontroladores. Disponível em: <<https://pictoblox.ai/>>. Acesso em 12/2021.

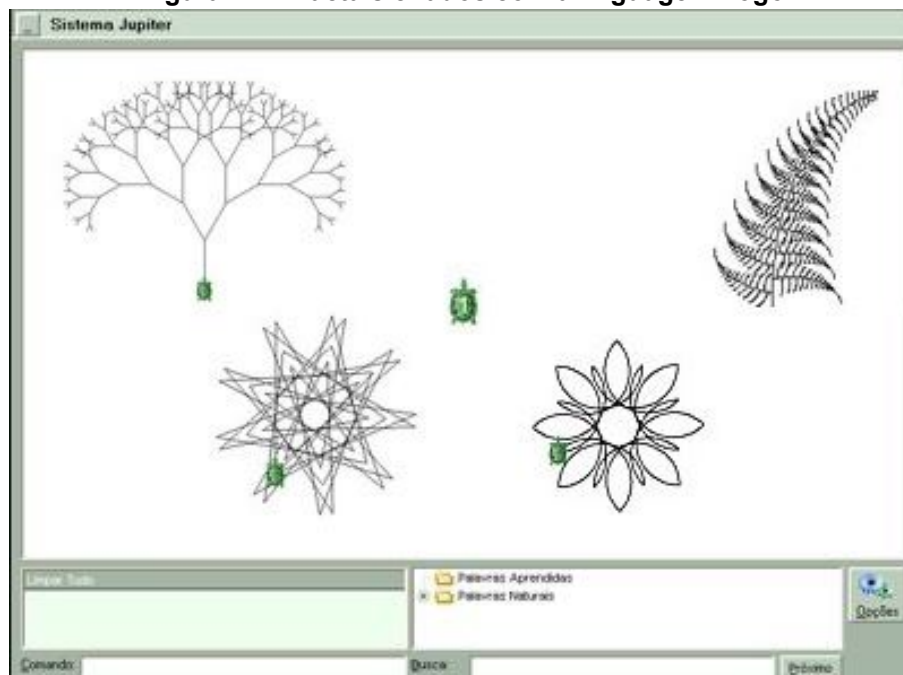
#### 4.5.5 X-Logo

X-Logo, Figura 27, é uma linguagem de programação, isto é, um meio de comunicação entre o computador e a pessoa que irá usá-lo. A principal diferença entre Logo e outras linguagens de programação está no fato de que esta foi desenvolvida para ser usada por crianças e para que as crianças possam, com ela, aprender outras coisas. Deste modo, **Logo** é uma linguagem simples e poderosa. Simples porque é fácil de aprender, afinal, pessoas alfabetizadas, de qualquer idade, podem programar em seu primeiro contato com ela, sem necessitar que tenham muitos conhecimentos prévios. E poderosa porque possui recursos sofisticados, que atendem às exigências de programadores mais experientes.

**Para saber mais:**

- <https://projetologo.neocities.org/texto1.html>
- [https://youtu.be/3n8cZAQKOVw?list=PL838IdaPZmcuJuP-fe\\_9WbXVgOMPWEd00](https://youtu.be/3n8cZAQKOVw?list=PL838IdaPZmcuJuP-fe_9WbXVgOMPWEd00)
- <https://projetologo.webs.com/xlogo.html>
- <https://xlogo.tuxfamily.org/pt/examples/potencia.html>
- <https://www.nied.unicamp.br/projeto/super-logo/>

**Figura 27 - Fractais criados com a linguagem Logo**



Fonte: Site Dia a dia educação<sup>33</sup>.

<sup>33</sup> Disponível em: <<http://www.gestaoescolar.diaadia.pr.gov.br/arquivos/File/tutoriais/xlogo.pdf>>. Acesso em 05/2022.

#### 4.5.6 HTML

O código HTML, uma abreviação de Hyper Text Markup Language, ou, em português, Linguagem de Marcação de Hipertexto, é uma das mais utilizadas para escrever páginas Web. Desta forma, quando você utiliza um navegador de internet, muito provavelmente ele esteja usando de linguagem html, afinal, é ela que geralmente está por trás de toda engenharia de software.

Porém, como se trata de uma linguagem para desenvolver páginas de documentos eletrônicos para a internet, o HTML não é em si uma linguagem de programação, já que seu objetivo não é criar um software, mas marcar a estrutura de linguagens de programação, o posicionamento dos elementos e a divisão de uma página em blocos visuais, além de distribuir imagens, vídeos e textos combinando os elementos entre si.

##### **Para saber mais:**

- Apostila Fundamentos do Desenvolvimento Web, de Keila Brito.

<http://proedu.rnp.br/handle/123456789/703>

- Curso completo e tutorial de HTML:

<http://www.nce.ufrj.br/ginape/cursohtml/conteudo/sumario.htm>

<https://www.w3schools.com/html/>

#### 4.5.7 Pascal

**Pascal** é uma linguagem de programação estruturada, ou seja, é uma forma de programação de computadores que preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas: **sequência, decisão e interação**. Assim, a programação estruturada orienta os programadores para a criação de estruturas simples em seus programas, usando as sub-rotinas e as funções.

Embora não tenha sido a primeira linguagem a incorporar conceitos da programação estruturada, Pascal é considerado um marco na evolução das linguagens de programação devido a suas características e às várias inovações introduzidas na sua época, como, por exemplo, a utilização de procedimentos que leem e escrevem campos individuais, em vez de registros completos, e o uso de funções e procedimentos recursivos, dentre outras.

Esta linguagem foi desenvolvida entre os anos de 1968 e 1970 pelo professor Niklaus Wirth, na Universidade Técnica de Zurique, na Suíça. Seu nome é uma homenagem a Blaise Pascal (1623-1662), matemático e filósofo francês que inventou a primeira calculadora mecânica.

#### 4.5.8 Linguagem C

A linguagem **C** é descrita como uma das linguagens de programação mais antigas, mantendo-se relevante até os dias atuais, por esse motivo é considerada a linguagem mãe, dando origem a diversas outras linguagens. Criada em 1972 por Dennis Ritchie e Ken Thompson, a linguagem C tinha o propósito inicial de ser usada no desenvolvimento de uma versão do sistema operacional Unix, e atualmente tem diversas aplicações, inclusive na robótica. Sendo assim, muitas das nossas oficinas didáticas serão desenvolvidas utilizando a linguagem C, para podermos nos familiarizar com essa linguagem.

A linguagem C é chamada de linguagem nível intermediária, visto que combina os elementos das linguagens de alto nível com o funcionalismo da linguagem nível máquina, sendo esta, utilizada na programação dos computadores e máquinas derivadas e, em última instância, na comunicação com estas máquinas.

Com a linguagem C é possível programar microcontroladores, sistemas eletrônicos físicos, que se assemelha a um computador em tamanho reduzido, como o Arduino, assunto de um próximo capítulo, usado em sistemas embarcados<sup>34</sup>, que são dispositivos com capacidade de processamento de dados e que estão inseridos em um determinado dispositivo ou produto, para desempenhar uma função ou servir a uma aplicação específica.

Programar em C, permite entender os processos ocultos que acontecem em nossas máquinas e, é uma ótima linguagem para programadores iniciantes. Um programa em C é composto por um conjunto de funções, sendo uma delas a função *main* pela qual o programa começa a ser executado. Após cada comando em C deve-se colocar um “ ; ” (ponto-e-vírgula). Um programa em C deve ser indentado (reco, neologismo derivado da palavra em inglês indented), facilitando sua leitura.

---

<sup>34</sup> Disponível em: <<https://www.embarcados.com.br/arduino-leonardo/>>. Acesso em 06/2023.

## 5. Robótica Educacional, Arduino e componentes eletrônicos

*“Uma máquina consegue fazer o trabalho de 50 homens ordinários. Nenhuma máquina consegue fazer o trabalho de um homem extraordinário.”*  
**Elbert Hubbard**

Esse capítulo tem por objetivo apresentar alguns conceitos e definições sobre a temática Robótica Educacional com Arduino, assim como, propor formas de aplicar em sala de aula o desenvolvimento da robótica educacional, com a utilização da placa Arduino e também o emulador de Arduino. Os componentes eletrônicos básicos utilizados, tanto como a metodologia utilizada no desenvolvimento das oficinas didáticas estão diretamente relacionadas conforme a pesquisa bibliográfica e também, durante a pesquisa exploratória ao utilizar o Tinkercad e sua biblioteca de componentes de maneira a reproduzir tais experimentos com o uso da ferramenta e a partir daí, simular o comportamento dos circuitos eletrônicos de modo teórico-prático.

### 5.1 A Robótica

A robótica é a ciência que estuda a tecnologia em máquinas, computadores, softwares e sistemas, associadas a concepção e construção de robôs. Como o próprio nome sugere, é a ciência e o estudo de robôs que são mecanismos automáticos que utilizam circuitos integrados para realizarem atividades e movimentos humanos simples ou complexos, que interagem com a realidade. Atualmente, segundo o dicionário Michaelis, “robótica é ciência e técnica que envolve a criação, a construção e a utilização de robôs” (ROBÓTICA, 2022). Em suma, robótica é um ramo da tecnologia que engloba: Construção, Mecânica, Eletrônica e Programação.

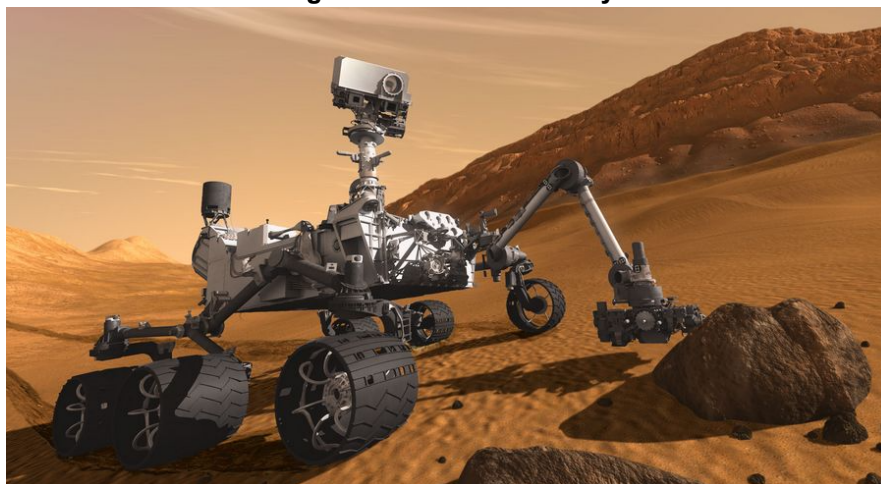
Pela história antiga, acreditava-se que os gregos e romanos já desenvolviam diferentes tipos de máquinas capazes de realizar movimentos automatizados no século X a.C. na dinastia Zhou Ocidental. Um humanóide que podia cantar e dançar foi construído pelo artesão Yan Shi e em Kaifeng na China, no ano de 1088, foi construída por Su Song uma torre de relógio de 10 metros, o Cosmic Engine.

No século IV a.C., o matemático Archytas de Tarentum postulou um pássaro mecânico que ele chamou de “O Pombo”, que era movido a vapor. Já René Descartes, século XVII, acreditava que animais e seres humanos eram máquinas biológicas, e não é para menos, pois o corpo humano é uma máquina perfeita, já comprovado por vários estudos científicos.

“Um robô é um sistema autônomo que existe no ambiente físico mundo, pode sentir seu ambiente e pode agir sobre ele para alcançar alguns objetivos.” (MATARIĆ, 2007, p.20).

A palavra “robô” vem da palavra eslava “rabota”, que significa “trabalho”. Esta palavra apareceu pela primeira vez na peça R.U.R. “Robôs Universais de Rossum”, do dramaturgo checo Karel Čapek, em 1920. Os robôs e seus nomes já são diversos, tem se perpetuado, ganhando espaço e na atualidade já temos robôs que atuam nas diversas áreas de trabalhos com os mais variados tipos e modelos que se adequam à necessidade com modelos customizados para ganhar tempo e espaço. Por falar em espaço, a NASA Administração Nacional da Aeronáutica e Espaço, uma agência do governo federal dos Estados Unidos responsável pela pesquisa e desenvolvimento de tecnologias e programas de exploração espacial, criou o mais famoso robô Curiosity, Figura 28, para uma missão espacial de exploração do planeta Marte com data de lançamento em 2011, que foi um programa considerado um sucesso com muitos avanços.

**Figura 28 - Robô Curiosity**



Fonte: Página da NASA<sup>35</sup>.

<sup>35</sup> NASA, disponível em: <<https://mars.nasa.gov/msl/home/>>. Acesso em 03/2022.

Esses avanços não só na área espacial, mas, nas diversas áreas e setores têm exigido cada vez mais dos trabalhadores e a capacitação dos profissionais é fundamental nesse sentido, pois, muitos não correspondem mais às exigências do trabalho e mercado diante todas as mudanças da atualidade no meio profissional e social. Algumas das vantagens da robótica são:

- Inovação - O trabalho com robótica exige novas ideias, estimula a criatividade, o planejamento, a execução de projetos complexos e a valorização dessas diversas etapas.
- Empreendedorismo - O profissional que trabalha com robótica sempre é estimulado a criar novas possibilidades, desenvolver novas alternativas, precisa vender suas ideias e soluções para conseguirem apresentar soluções que funcionem e sejam vantajosas a curto, médio e longo prazo, dependendo de cada projeto.
- Novas oportunidades - O espírito investigativo, a cooperação e tomada de decisões, faz parte das novas tendências e estimula o conhecimento, a busca pela ciência, a tecnologia e a matemática. É um excelente caminho para fomentar novas gerações de pesquisadores, capazes de oferecer soluções para problemas ainda mais complexos.
- Maior produtividade - O aprendizado da programação e da automação estimula as pessoas a organizarem melhor suas ações e isso vai refletir na capacidade de organização de modo geral.
- Melhoria na gestão - A gestão fica facilitada, pois os robôs atuam a partir de base de dados e sistemas integrados que geram informações instantâneas, podendo ser melhor visualizadas e compreendidas.

Além dessas e outras vantagens, robôs podem realizar tarefas que humanos dificilmente conseguiriam, podem ainda ser utilizados em atividades insalubres e perigosas, o que permite aos funcionários que as tarefas de coordenação do trabalho robótico sejam de um local seguro, diminuindo ou eliminando os riscos para a saúde do trabalhador.



Na atualidade a robótica avançada tem ganhado espaço e ajudado na otimização do tempo, no aumento da produtividade, no aumento da margem de lucros dos negócios, entre outras vantagens advindas da expansão das novas tecnologias de automação e com a chamada indústria 4.0, que é equivalente a 4º Revolução Industrial, na qual a tecnologia é utilizada para mediar boa parte dos processos de produção, deixando-os mais ágeis e modernos e têm impulsionado mudanças de hábitos e de comportamentos da sociedade contemporânea.

Nessa perspectiva, o uso da tecnologia na educação é uma necessidade que não podemos dispensar, todo profissional atualizado em estudos e desenvolvimentos entende que a tecnologia pode facilitar a educação desde que usada corretamente, e a robótica já é uma realidade e faz-se necessário esse conhecimento para os estudantes, futuros profissionais. Ao se familiarizar com as tendências relacionadas à tecnologia, os professores entrarão em contato com novos métodos de ensinar. Com isso, ganha-se familiaridade com os diversos conteúdos, aumentando a capacidade dos profissionais, com as tendências relacionadas a tecnologia na educação, assim os professores poderão desenvolver o hábito de continuar se familiarizando com os novos programas, ferramentas, aplicativos de ensino, etc.

Nas instituições de ensino a tecnologia tem se tornado uma aliada com fundamental importância. Nesse contexto, a programação e a robótica, mais especificamente a robótica educacional, trazem aspectos diferenciados e possibilitam ganhos para os processos de ensino e aprendizagem.

Não penso que todas as pessoas tenham que ter um alto grau de proficiência, acredito que as pessoas precisam lidar bem com esse novo mundo.

Mas, quando nós, professores, iniciamos nossos estudos sobre os robôs e as formas de se trabalhar essa temática, muita são as dúvidas, principalmente no que diz respeito aos elementos e componentes básicos dos robôs, e pensando nisso segue abaixo alguns dos principais:

- |  |  |
|--|--|
| <input type="checkbox"/> Suprimentos de energia. Fonte de alimentação. | <input type="checkbox"/> Servo Motors.             |
| <input type="checkbox"/> Sensores.                                     | <input type="checkbox"/> Manipuladores.            |
| <input type="checkbox"/> Motores elétricos CA/CC.                      | <input type="checkbox"/> Efetores Finais.          |
| <input type="checkbox"/> Mecanismo de condução.                        | <input type="checkbox"/> Dispositivos de locomoção |

É fundamental conhecermos um pouco sobre cada um desses componentes e quais são as suas funcionalidades, mas, isso não significa que é preciso sermos especialistas no assunto e sim termos um mínimo de conhecimento para podermos desenvolver nossos projetos com maior facilidade e eficiência.

## 5.2 Robótica educacional

**Robótica educacional** é algo bastante falado, mas, ainda muito pouco trabalhado nas escolas de modo geral. Os motivos são diversos, que vão desde a falta de recursos até a falta de formação e informações principalmente. Trabalhar com robótica tem trazido muita satisfação para quem optou por essa escolha e, corroborando com essa afirmação, “Poucas pessoas ficam entediadas com robôs que podem programar e controlar, porque o desafio é aberto e interminável. (Matarić 2007, p. 302).

A robótica educacional é também designada por robótica pedagógica, é um termo que caracteriza uma ferramenta pedagógica de aprendizagem baseada na construção de dispositivos controláveis por softwares e computadores, programados a partir de modelos multidisciplinares, pois nela é aplicado o conhecimento de diversas áreas. Esse termo não é algo novo, surgiu por volta de 1960, em seguida Seymour Papert publica vários livros na década de 90, desenvolvendo sua teoria sobre o construcionismo com objetivo de criar ambientes interessantes e utilizar “objetos-de-pensar” baseado no aprender pelo fazer, tendo resultado na linguagem Logo<sup>36</sup> um trabalho diferenciado para implantar a informática na educação.

“As crianças amam construir coisas, então escolhemos um conjunto de construção e a ele acrescentamos o que quer que seja necessário para torná-lo um modelo cibernético. Elas deveriam ser capazes de construir uma tartaruga com motores e sensores e ter uma forma de escrever programas em LOGO para guiá-las; ou, se desejassem fazer um dragão, um caminhão ou uma cama-despertador, deveriam ter essa opção também. Elas seriam limitadas apenas por suas imaginações e habilidades técnicas.” (PAPERT, 2008, p. 184).

---

<sup>36</sup> Projeto Logo, foi desenvolvido por Seymour Papert nos anos 60. Logo é uma linguagem de programação. Disponível em: <<https://projetologo.neocities.org/texto1.html>>. Acesso em 07/2023.

A robótica educacional é uma metodologia proposta para o desenvolvimento das habilidades do Pensamento Computacional, com objetivo de estimular o espírito investigativo que compõem o currículo oferecido pelas escolas na educação básica. A robótica pedagógica, é uma metodologia que propõe desafios por meio de projetos lúdicos, visa levar o estudante a questionar, pensar e procurar soluções, a sair da teoria para a prática usando ensinamentos obtidos em sala de aula, na vivência cotidiana, nos relacionamentos, conceitos e valores. Nesse contexto, valoriza o trabalho em grupo, a cooperação, promove o diálogo, e tudo isso inserido em um processo de construção e reconstrução.

A robótica está sendo conhecida por ser uma ferramenta que estimula os estudantes a aprenderem na prática o funcionamento das máquinas e ferramentas. É a ciência que estuda as tecnologias associadas a este estímulo dentro de sala de aula e, resulta em alguns benefícios que podemos destacar:

- Estimula a criatividade;
- Aprimora o pensamento matemático;
- Desenvolve o raciocínio lógico;
- Desenvolve a criticidade;
- Melhora as habilidades para solução de problemas;
- Cria argumentação e mecanismos cognitivos para compreender fenômenos;
- Transforma o processo de ensino e aprendizagem em uma experiência divertida;
- Desenvolve o protagonismo;
- Incentiva o trabalho em grupo.

A partir desses benefícios da robótica educacional o estudante inicia um processo de busca pelo conhecimento da capacidade analítica de observação de fenômenos, desenvolvendo assim habilidades específicas e gerais que beneficiam seu desenvolvimento social, pessoal e profissional. De acordo com Oliveira (2013):

“A Implantação da robótica educativa nas escolas visa proporcionar aos alunos e ao professor a construção e desenvolvimento do pensamento de cada aluno, procura despertar o raciocínio lógico, a criatividade, a autonomia no aprendizado, a compreensão de conceitos e procura melhorar a convivência em grupo, tratar a cooperação, o planejamento de atividades e tarefas.” (OLIVEIRA 2013, p.4).

A robótica educacional é uma ferramenta com muito potencial, visto que proporciona possibilidades, desperta e transforma a criatividade e o interesse dos estudantes para a descoberta e o protagonismo, dentro das perspectivas matemáticas, é possível fazer uso, desde a concepção da ideia, de algoritmos, estabelecendo padrões e procedimentos essenciais para construção do conhecimento, passando pela confecção do objeto, a montagem, até sua programação.

“A utilização da robótica em sala de aula pode se configurar em uma proposta privilegiada para instigar e desafiar os alunos, ampliando o aprendizado de matemática e diminuindo suas dificuldades de aprendizagem.” (LIMA, 2020, p.1).

A utilização da robótica educacional possibilita que inúmeras capacidades, tanto do instrutor como do usuário, sejam desenvolvidas, desde a melhora no poder de solucionar problemas até mesmo o desenvolvimento de trabalhos de forma colaborativa. Norma Godoy (1997 *apud* CASTRO, 2008, p.) apresenta uma classificação dos principais objetivos da robótica educacional, detalhando-os em:

**Objetivos Gerais:** construção dos protótipos com motores e sensores, adaptando elementos dinâmicos como engrenagens, rodas, dentre outros.

**Objetivos Psicomotores:** desenvolver a motricidade, proporcionar a formação de habilidades manuais, desenvolver a concentração e a observação, motivar a precisão de seus projetos.

**Objetivos Cognitivos:** estimular a aplicação das teorias formuladas às atividades concretas, desenvolver a criatividade dos alunos, analisar e entender o funcionamento dos mais diversos mecanismos físicos.

**Objetivos Afetivos:** promover atividades que gerem cooperação em trabalhos de grupo e estimular o crescimento individual através da troca de projetos e ideias. A partir dos objetivos destacados, pode-se inferir que a elaboração e prática de trabalhos com robótica educacional, por meio de um planejamento adequado, permitem que esses objetivos sejam alcançados de forma satisfatória.

Os benefícios da robótica educacional são diversos e os objetivos de sua utilização vão desde a facilitação do desenvolvimento das habilidades do pensamento computacional ao aumento de interesse em frequentar as aulas por exigir maior interação e espírito de equipe entre os estudantes que trabalham em grupo e apresentam também maior protagonismo no desenvolvimento dos projetos que geralmente são desenvolvidos de forma interdisciplinar. Almeida (2007) corrobora com essa afirmação:

“O principal objetivo da robótica educacional é promover ao educando o estudo de conceitos multidisciplinares, como física, matemática, geografia, artes, biologia, entre outros. Há variações no modo de aplicação e interação entre os alunos, estimulando a criatividade e a inteligência e promovendo a interdisciplinaridade.” (ALMEIDA, 2007, p.2).

A interdisciplinaridade é uma das melhores formas de se trabalhar a robótica educacional com o desenvolvimento de uma metodologia diferenciada que possibilita articulação entre os conteúdos nas diversas áreas do conhecimento, do desenvolvimento das várias habilidades, facilita a compreensão dos diversos conteúdos, também possibilita o desenvolvimento do trabalho colaborativo, o raciocínio lógico e a criatividade. Silva (2009, p. 45), em seus estudos, salienta que a utilização da Robótica Educacional tem como objetivos:

- desenvolver a autonomia, isto é, a capacidade de se posicionar, elaborar projetos pessoais, participar na tomada de decisões coletivas;
- desenvolver a capacidade de trabalhar em grupo: respeito a opiniões dos outros;
- proporcionar o desenvolvimento de projetos utilizando conhecimento de diversas áreas;
- desenvolver a capacidade de pensar múltiplas alternativas para a solução de um problema;
- desenvolver habilidades e competências ligadas à lógica, noção espacial, pensamento matemático, trabalho em grupo, organização e planejamento de projetos envolvendo robôs;

- promover a interdisciplinaridade, favorecendo a integração de conceitos de diversas áreas, tais como: linguagem, matemática, física, ciências, história, geografia, artes, etc.

Na Robótica Educacional o estudante encontra motivação, e ao trabalhar em grupo discute soluções para problemas cotidianos e se torna protagonista de seu conhecimento. Ainda, aprende o uso e a criação de tecnologias com base nos três pilares da robótica: mecânica, elétrica e programação. Assim, o estudante percebe que pode usar conhecimentos adquiridos da robótica em outras disciplinas (matemática, história, física, geografia, etc.) sentindo-se motivado a empenhar-se mais nos estudos, desta forma, tendo como resultados o aumento no rendimento, com facilidade de fixação dos conceitos aprendidos em sala de aula de forma divertida e descontraída. O diferencial está na autonomia, na responsabilidade, no senso crítico e na maior independência em criar soluções para os novos desafios do mundo atual.

No entanto, a falta de recursos sempre foi um complicador, isso pode ser um desafio para os educadores que desejam levar a robótica para a sala de aula. Mas, a prática também pode ser feita com materiais do cotidiano, materiais reutilizáveis entre outros de fácil acesso.

### **5.3 Scratch**

O Scratch é um projeto do grupo Lifelong Kindergarten, uma comunidade online gratuita com uma linguagem de programação visual pela qual qualquer pessoa pode criar suas próprias histórias, jogos e animações.

O Scratch promove o desenvolvimento das habilidades do PC, possibilitando o ensino e aprendizagem criativos, a resolução de problemas, auto expressão e colaboração:

“O Scratch possui personagens, também chamados de sprites, em que o principal é um gato – embora seja possível trocar por outro de livre escolha. É ele que executa os movimentos, as imagens, as animações, os sons, as histórias, os jogos.” (FERREIRA; DE OLIVEIRA, 2018, p. 83).

O mascote do Scratch geralmente é um gatinho, como na Figura 29. Esta comunidade possui vários outros personagens e tem-se aperfeiçoada de forma que é possível trabalhar os projetos de robótica educacional de maneira facilitada pela linguagem visual em blocos do Scratch e, a mesma adapta-se com as mais diversas placas controladoras e ambientes operacionais da atualidade.

Figura 29 - Scratch, mascote e blocos



Fonte: Autoria própria, 2022.

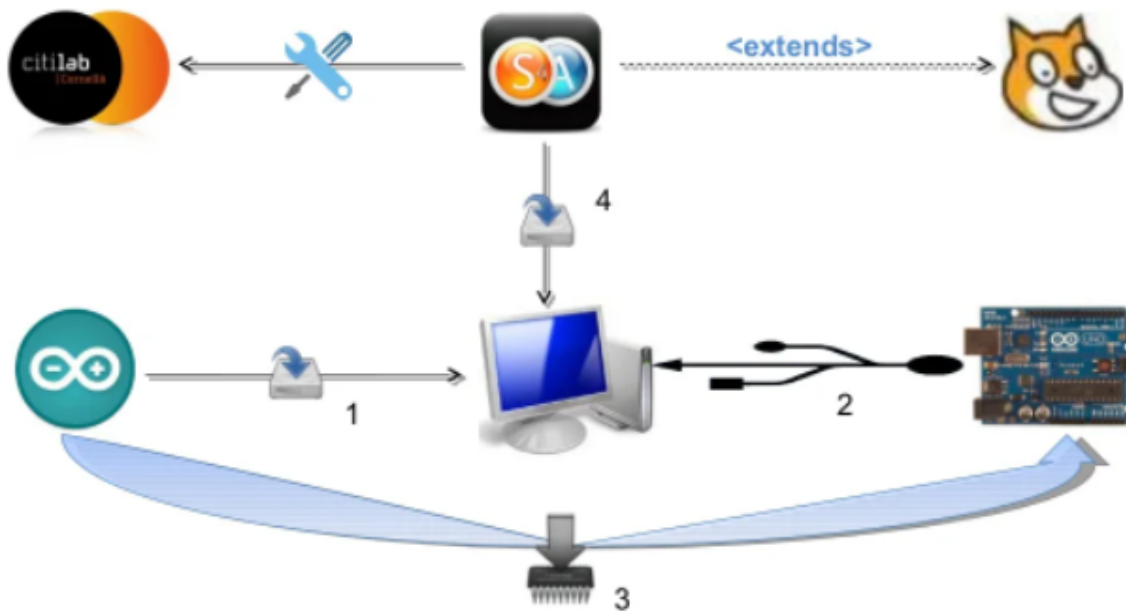
Aprender a programar e construir projetos ficou mais fácil com a ferramenta Scratch e suas possibilidades para todos os públicos, desde aqueles que não exigem conhecimentos em programação até os mais avançados que requerem um pouco mais de dedicação do usuário. Beer (2013), corrobora com essa afirmação:

“O Scratch ensina a lógica de programar um software ou um site, a mesma para qualquer linguagem de computador, da mais simples à mais complexa, que os cientistas de computação chamam de fluxo de controle.”(BEER, 2013, p. 88).

A interface da ferramenta Scratch, Figura 30, é bastante amigável e os estudantes, crianças, adolescentes e adultos têm apresentado apropriação satisfatória devido à facilidade de manuseio da mesma.

Figura 30 - Scratch e Arduino

## Usando Scratch e Arduino: Instalação



Fonte: Página Slideshare<sup>37</sup>.

### 5.4 Tinkercad

Dentre as diversas ferramentas digitais existentes, dispomos do Tinkercad, uma plataforma que permite construir projetos virtuais com **Arduino** e vê-los funcionando. Esse ambiente, Tinkercad, é um software gratuito de modelagem 2D e 3D (duas e três dimensões) da Autodesk, que pode ser usado inteiramente dentro de um navegador de internet. Disponível no site: <https://www.tinkercad.com/>.

O simulador Tinkercad, Figura 31, disponibiliza formas geométricas para construção de objetos de maneira simples e intuitiva. Na ferramenta de modelagem 3D, os estudantes poderão planejar, modelar e projetar os objetos desejados. Esses objetos devem ser seccionados e planejados para possibilitar a impressão no papel ou em uma cortadora a laser, caso a escola tenha à disposição.

<sup>37</sup> Slideshare. Disponível em: <<https://pt.slideshare.net/rafamachadoalves/o-uso-do-scratch-em-educacao-e-robotica>>. Acesso em 05/2022.



Figura 31 - Tinkercad



Fonte: Página Tinkercad<sup>38</sup>.

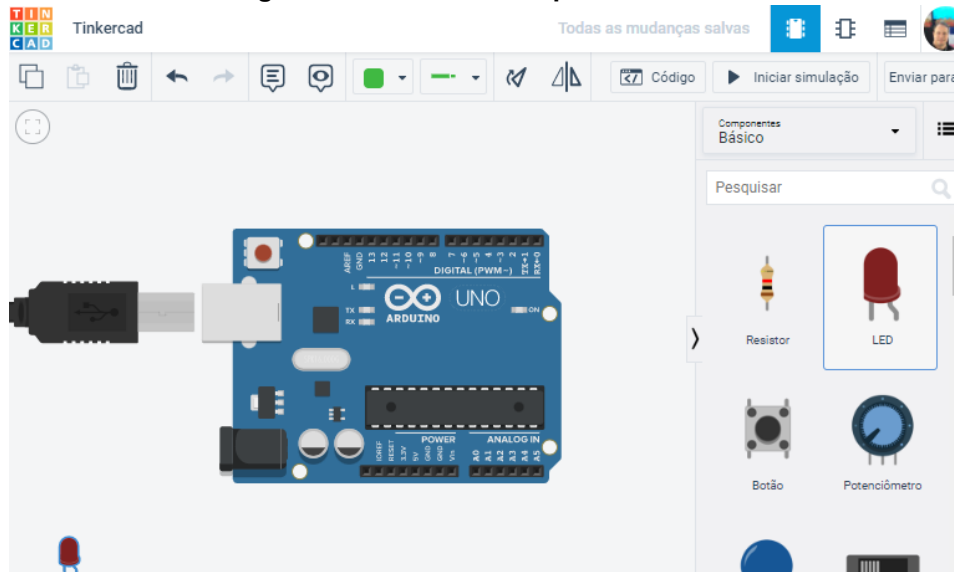
Sua interface disponibiliza de um ambiente amigável, intuitivo de fácil entendimento e permite criar projetos em eletrônica e codificação, também facilita muito a interação entre as mais diversas placas controladoras e ambientes operacionais da atualidade. Ainda disponibiliza lições para estudos e compreensão de circuitos eletrônicos, projetos diversos na área de robótica e, esse é o assunto principal desse capítulo, pois esse ambiente proporciona uma ampla aplicabilidade para nossos projetos dentro e fora de sala de aula, assim como podemos ver e construir alguns projetos por meio dessa ferramenta. Nesse sentido, Torres (2012) afirma que:

“Os simuladores computacionais são recursos tecnológicos que podem auxiliar em diversas áreas. No contexto abordado neste trabalho, os simuladores auxiliam os professores a provocar, favorecer e orientar situações controladas de ensino e aprendizagem, pedagogicamente interessantes para o nível de ensino que deseja. Do ponto de vista pedagógico, simuladores computacionais são softwares úteis e simples, uma vez que levam os alunos a interagirem com os mesmos, o que lhes facilita a compreensão de conceitos abstratos, leis e características do mundo físico.” (TORRES 2012, p.2).

Para realizar as simulações no Tinkercad, é importante conhecer um pouco do funcionamento e aplicação dos principais componentes elétricos, tais como: resistores, motores, sensores e LEDs. Enfim, adquirir um pouco de conhecimentos em eletrônica, e também do funcionamento da placa didática Arduino, Figura 32.

<sup>38</sup> Tinkercad. Disponível em: <<https://www.tinkercad.com/learn/circuits>>. Acesso em 05/2022.

**Figura 32 - Tinkercad e a placa Arduino**



Fonte: Autoria própria.

## 5.5 Eletrônica básica

Aprender eletrônica básica é fundamental, pois, a partir daí, é possível compreender como funcionam os principais componentes que veremos para nos preparar para as nossas tarefas diárias a desenvolver os projetos de robótica educacional. No entanto, você não precisa saber tudo sobre eletrônica e nem pense que esse assunto é complicado demais para você aprender e depois para ensinar outras pessoas. Para isso, se apropriar dos principais elementos que compõem uma placa eletrônica pode ajudar em tarefas posteriormente na sua vida Maker que nem sempre é fácil, mas tendo informações precisas pode auxiliar bastante.

Por conseguinte, conhecer alguns conceitos e componentes principais da eletrônica, os quais são fundamentais para quem estuda eletrônica e aplica nos projetos de robótica educacional:

“Ao empregar componentes eletrônicos no desenvolvimento de experimentos, mesmo dentro dos simuladores, a fim de evitar acidentes e o desperdício de componentes na execução do projeto final, é importante compreender como a eletricidade percorre os circuitos elétricos.” (SANCHES, 2020, p. 15).

Portanto, compreender os conceitos fundamentais da eletrônica, que é “o campo da ciência e da engenharia que estuda a forma de controlar a energia elétrica por meio de dispositivos e meios condutores ou semicondutores” (FUENTES, 2009, p. 13), é um passo importante para que durante o desenvolvimento e execução dos projetos de robótica educacional e de ensino maker tenhamos resultados satisfatórios. Já Beasant (1985) traz que:

“A eletrônica é o estudo do controle de pequenas correntes elétricas para fazer com que todos os tipos de equipamento eletronicamente funcionem [...] Graças a eletrônica podemos viajar, nos comunicar a nível mundial e universal, pois os astronautas estão explorando cada vez mais o espaço.” (BEASANT, 1985, p.04).

A eletrônica é o fio condutor dos elementos que compõem os projetos, ela se dedica a controlar a corrente elétrica para que ela faça algo útil, por exemplo, acender uma lâmpada, ligar um computador, etc.

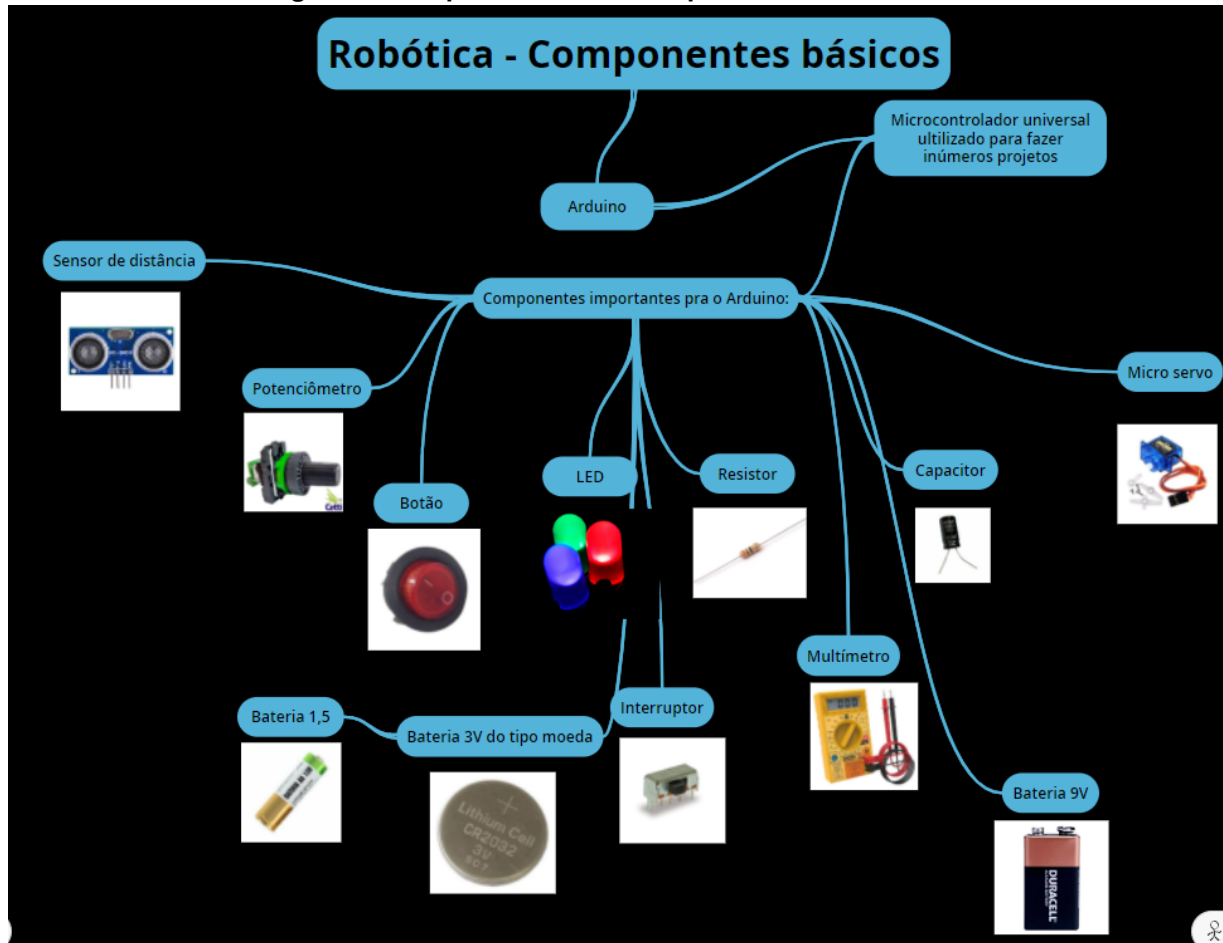
#### 5.5.1 Componentes eletrônicos básicos para a robótica

Os projetos de robótica educacional utilizam componentes eletrônicos que podem ser adquiridos em kits completos, ou separadamente em lojas e sites especializados em produtos de eletrônica. Para quem desejar, é possível trabalhar com componentes e materiais reaproveitáveis ou reutilizáveis, isso depende do planejamento e do projeto que se deseja construir.

“A melhor maneira de compreender a eletrônica é criar alguns circuitos simples para observar como funcionam os componentes...Existem dois tipos principais de eletrônica, a digital e a analógica. Os circuitos digitais utilizam pulsos de eletricidade, enquanto que os circuitos analógicos usam um fluxo de eletricidade que pode ter correntes e tensões continuamente variáveis.” (BEASANT, 1985, p. 26).

Na Figura 33, apresentamos alguns dos componentes básicos de eletrônica que podem ser utilizados na robótica, mas, vale lembrar que há uma gama enorme de materiais que podem ser utilizados, para desenvolvimento dos diversos projetos.

Figura 33 - Mapa mental dos componentes básicos da robótica



Fonte: Autoria própria pelo goconqr<sup>39</sup>.

### 5.5.2 Placas de circuitos eletrônicos

As placas de circuitos eletrônicos, também conhecidas como PCBs ou placas de circuito impresso, são compostas basicamente por placas de circuito como componentes centrais, são componentes eletrônicos que estão no centro da maioria dos projetos de engenharia eletrônica, desempenhando função chave na integração e comunicação entre as partes de um circuito. As placas tiveram evoluções significativas na atualidade, elas são de diversos tipos, tamanhos e modelos.

Você sabe identificar os principais elementos de uma placa eletrônica?

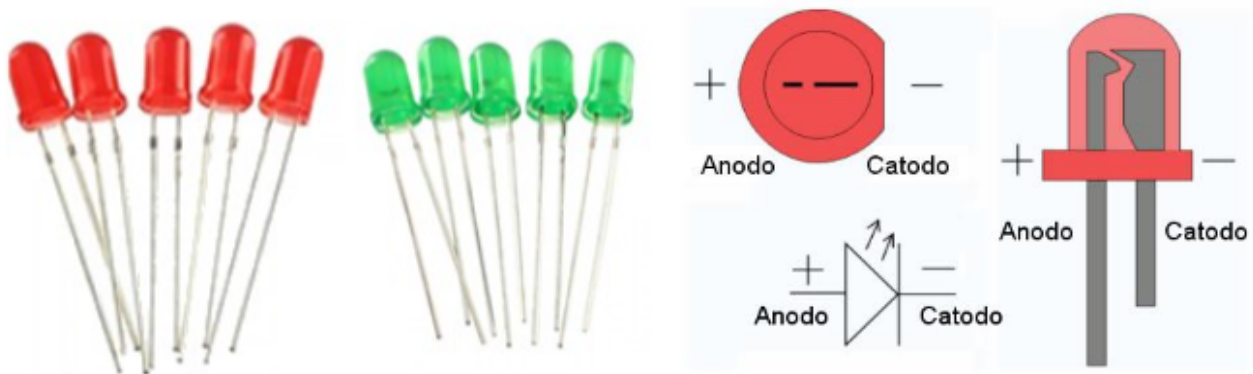
<sup>39</sup> Disponível em: <<https://www.goconqr.com/mapamental/36476439/robotica-componentes-basicos>>. Acesso em 03/2022.

Alguns elementos de uma Placa Eletrônica são: resistores, capacitores, indutores, sensores e antenas. São também conhecidos como componentes elétricos. Na maioria das placas eletrônicas existem pequenas letras que identificam os componentes que a compõem.

### 5.5.3 LED ou lâmpadas

O LED (L.E.D = Light Emitter Diode) é um componente eletrônico semicondutor, ou seja, um diodo emissor de luz, tem a propriedade de transformar energia elétrica em luz. Em suma, é uma lâmpada, um dispositivo elétrico que transforma energia elétrica em energia luminosa. Temos vários tipos de LEDs em variadas cores e ainda temos uma em especial que é o LED RGB, Figura 35, trata-se da associação de três LEDs distintos em um único LED, eles podem ser denominados de: catodo comum ou anodo comum, sendo o catodo comum o mais utilizado, e para identificar o terminal que equivale ao comum, basta ver qual terminal é o maior, ou ver qual está conectado a maior “placa” presente no interior do LED, como é possível ver na ilustração, Figura 34.

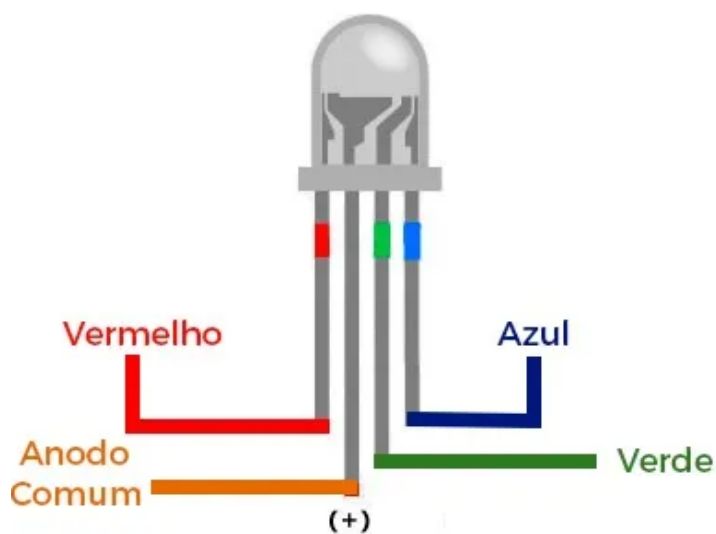
**Figura 34 - LED e seus principais elementos**



Fonte: Página Vida de Silício<sup>40</sup>.

<sup>40</sup> Disponível em: <<https://portal.vidadesilicio.com.br/como-utilizar-o-led-rgb-com-arduino/>>. Acesso em 05/2022.

**Figura 35 - LED RGB**

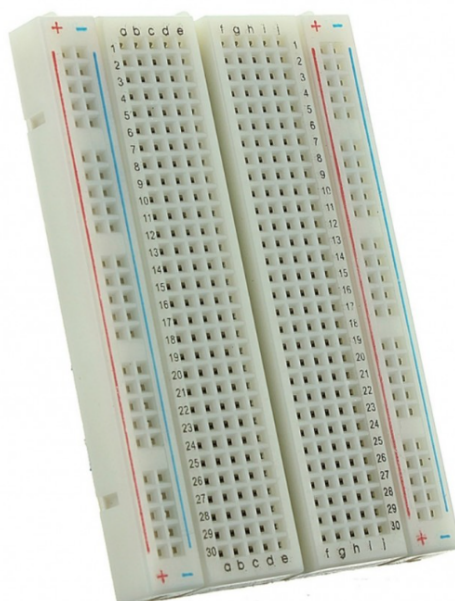


Fonte: Página Vida de Silício.

#### 5.5.4 Placa protoboard

Protoboard ou matriz de contato, como na Figura 36, é uma placa com diversos furos e conexões condutoras verticais e horizontais para a montagem de circuitos elétricos experimentais. Seu uso tem a vantagem de aproveitamento em vários projetos por dispensar a soldagem.

**Figura 36 - Placa Protoboard**



Fonte: Autoria própria.

### 5.5.5 Resistores

Resistores, Figura 37, são componentes eletrônicos que resistem à passagem de corrente elétrica. Um resistor ou uma resistência é um dispositivo elétrico muito utilizado em eletrônica, ora para transformar energia elétrica em energia térmica por meio do efeito joule, ora para limitar a corrente elétrica em um circuito. Existe uma fórmula para calcular o resistor adequado para um LED, por exemplo, não vamos entrar em maiores detalhes, pois não é o nosso objetivo, segue a fórmula:

$$R = (V_{\text{alimentação}} - V_{\text{LED}}) / I .$$

**R** é a resistência em ohms do resistor adequado para o LED, isso é o que você quer descobrir.

**V<sub>alimentação</sub>** é a tensão em volts da fonte de alimentação que você vai usar no LED.

**V<sub>LED</sub>** é a tensão em volts do LED.

**I** é a corrente do LED em amperes.

**Figura 37 - Resistores**



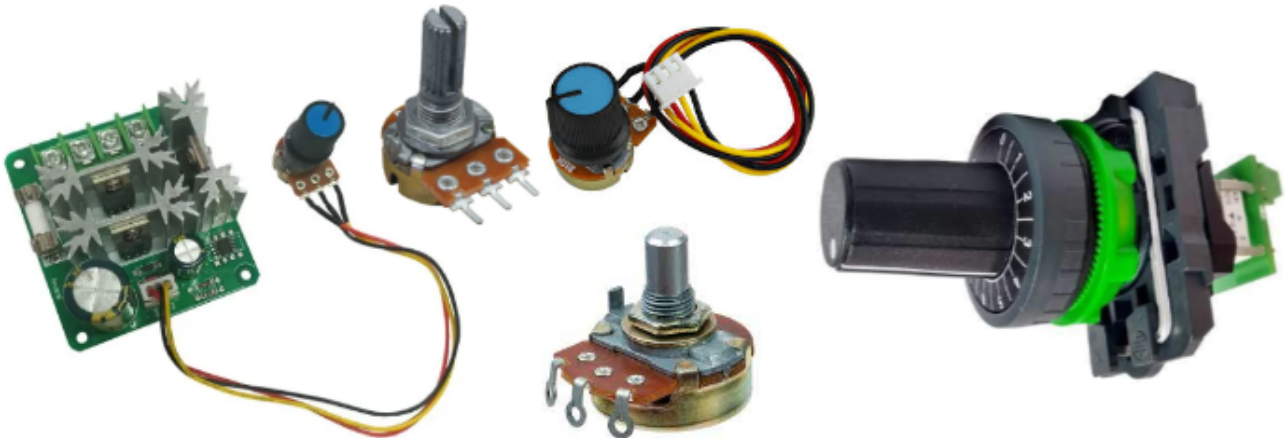
Fonte: Autoria própria.

### 5.5.6 Potenciômetro

Potenciômetro, Figura 38, é um componente eletrônico que possui resistência elétrica ajustável, normalmente tem três terminais onde a conexão central é deslizante e manipulável via cursor móvel. Se todos os três terminais são usados, ele atua como um divisor de tensão. Quando aplicamos uma tensão elétrica entre os terminais de início e fim, a tensão elétrica obtida no terminal central irá variar conforme o cursor se move ao longo da pista.

Os potenciômetros e os resistores têm essa finalidade de limitar o fluxo de corrente elétrica em um circuito, a diferença é que o potenciômetro pode ter sua resistência ajustada e o resistor comum não pode, pois, ele possui um valor de resistência fixo.

**Figura 38 - Potenciômetros**



Fonte: Autoria própria.

### 5.5.7 Motor de passo x motor DC

Um motor elétrico é um dispositivo eletromecânico que converte energia elétrica em energia mecânica. Os motores de passo que podem ser rotacionados livremente, também chamado de motor síncrono AC (Corrente Alternada), transforma a energia elétrica inserida em seus pólos em energia mecânica através da rotação de seu eixo, são de corrente contínua e possuem várias bobinas e são usados em aplicações que exigem uma alta precisão, por isso seu torque é muito baixo, se movem com incrementos ou passos que podem ser quantificados. A velocidade de um motor AC é controlada através da variação da frequência, enquanto a velocidade de um motor DC (Corrente Contínua) é controlada através da variação da corrente, pode ser alimentado com até 6 volts. Os motores de corrente contínua (também conhecidos por motores DC, Figura 40, ou CC, Figura 39) são igualmente muito comuns no setor industrial, pois apresentam vantagens consideráveis, dependendo das suas características construtivas.

Outro tipo de motor para comparar a esses dois é o servo motor que é muito utilizado em aplicações de robótica.



O eixo de um servo motor não costuma ter tanta liberdade em seus movimentos quanto ao motor de passo, geralmente é de apenas 180°. Ele é basicamente um motor que podemos controlar a sua posição angular por meio de um sinal PWM (Pulse Width Modulation), ou modulação por largura de pulsos, utilizado para posicionar e manter um objeto em uma determinada posição. O servo motor trabalha com servo-mecanismo que usa o feedback de posição para controlar a velocidade e a posição final do motor. Embora os servo motores não sejam uma classe específica de motor (podem ser CC ou CA, síncrono ou de indução), eles são destinados e projetados para uso em aplicações de controle de movimento que exigem posicionamento de alta precisão, reversão rápida e desempenho excepcional.

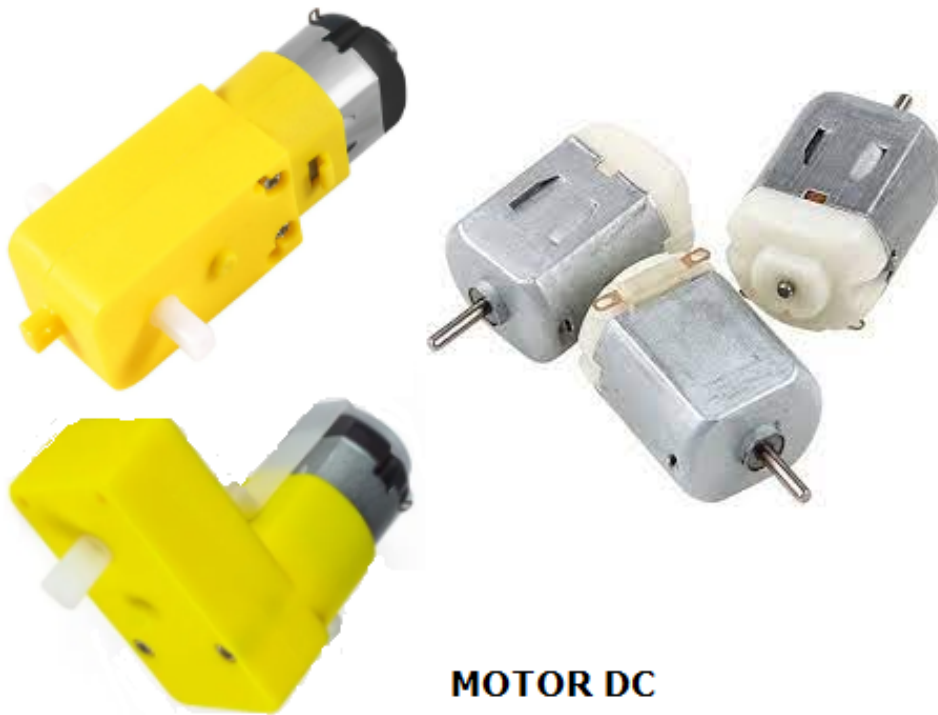
Figura 39 - Motores de corrente contínua



Fonte: Página Mundo da Elétrica<sup>41</sup>.

<sup>41</sup> Disponível em: <<https://www.mundodaeletrica.com.br/motor-de-corrente-continua-caracteristicas-e-aplicacoes/>>. Acesso em 05/2022.

**Figura 40 - Motor DC**



## **MOTOR DC**

Fonte: Aatoria própria.

### 5.5.8 Jumpers

Os jumpers, Figura 41, são utilizados em placas e alguns dispositivos como chaves elétricas, por exemplo, para ativar discos rígidos, regular ou desativar funções específicas do sistema que não são acessíveis via software. Os Jumpers arduino são pequenos fios condutores que podem ser conectados a uma protoboard para interligar dois pontos do circuito em projetos eletrônicos. Esses jumpers podem ser confeccionados com pedacinhos de fios com materiais reutilizáveis.

**Figura 41 - Jumpers**



Fonte: Aatoria própria.

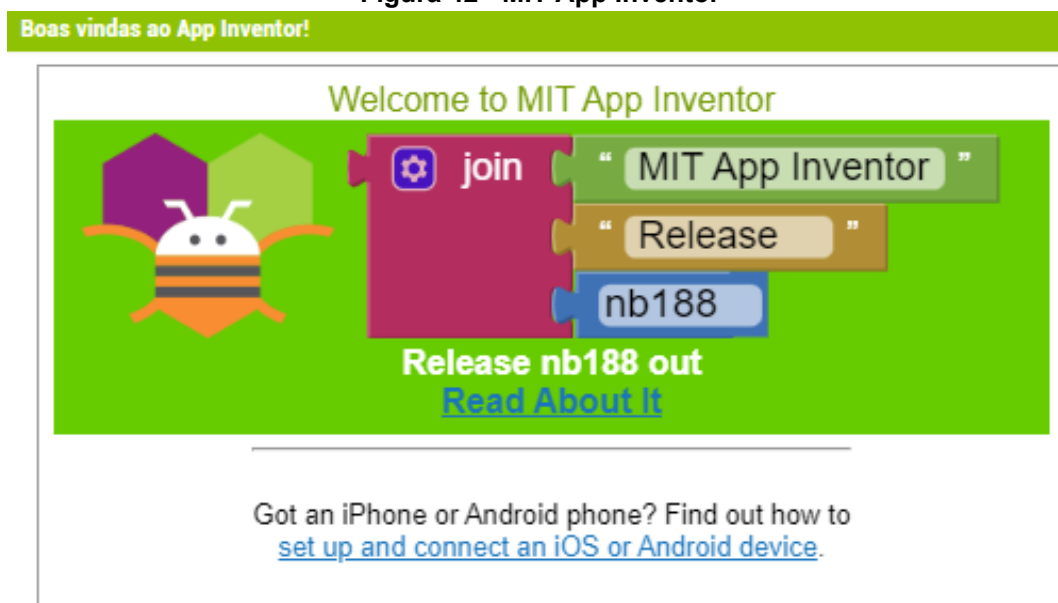
## 5.6 MIT APP Inventor

Já pensou em criar um aplicativo do seu jeito? Com as funcionalidades que você deseja? O App Inventor, Figura 42, é uma ferramenta importante para o desenvolvimento de aplicativos, os quais podemos construir pensando na aplicabilidade para os nossos projetos em robótica ou até mesmo para o nosso dia a dia. A interface do App Inventor é organizada, limpa e visa uma otimização do trabalho do usuário com a utilização do clique e arraste para inserir o item desejado no projeto. Segundo Cordeiro (2017), do Blog AndroidPro:

“Google App Inventor é uma ferramenta de programação baseada em blocos lógicos, desenvolvida pela Google, que permite a criação de aplicativos para smartphones que rodam no sistema operacional Android, sem que seja necessário conhecimentos profundos de programação.” CORDEIRO (2017, p. 13).

O site para acesso a esta ferramenta está disponível no endereço (<https://appinventor.mit.edu/>), basta acessar e realizar o cadastro para começar a criar e construir e testar os aplicativos.

Figura 42 - MIT App Inventor

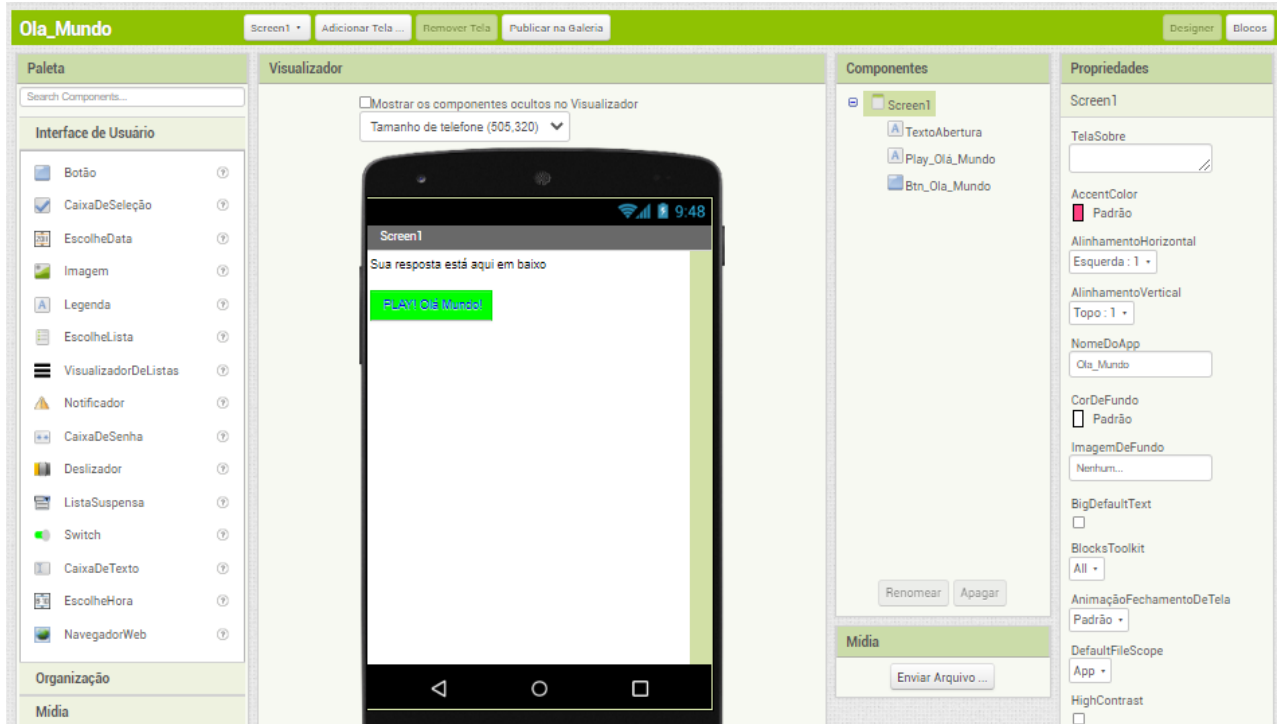


Fonte: Página APP Inventor<sup>42</sup>.

<sup>42</sup> Disponível em: <<https://appinventor.mit.edu/>>. Acesso em 03/2022.

A interface do ambiente é bastante amigável e fácil de manusear como pode ser visto na Figura 43.

**Figura 43 - Área de trabalho do App Inventor**



Fonte: Autoria própria.

## 5.7 Arduino

O arduino é uma plataforma de prototipagem eletrônica open source, surgiu na Itália, foi projetada com objetivo de atender as demandas de projetistas amadores e estudantes, por isso foi pensada para ser uma ferramenta acessível e fácil de programar. É uma ferramenta utilizada para o ensino de robótica nos diversos níveis de ensino, pois é bastante acessível, fácil de configurar, montar e programar. As aplicações com a plataforma arduino são ilimitadas e vai depender muito da criatividade e da dedicação do desenvolvedor. Quando você conecta a placa ao computador, você pode escrever códigos para o Arduino através da linguagem C/C++.

O Arduino é composto por três elementos:

- A placa Arduino (hardware); (Objeto físico, imagem abaixo).
- Ambiente de programação; (<http://gg.gg/login-arduino-cc-criar-conta>).
- Comunidade. (<https://forum.arduino.cc/>).

Placa Arduino UNO, Figura 44 - um microcontrolador programável.

**Figura 44 - Placa Arduino UNO**



Fonte: Autoria própria.

Arduino é uma plataforma de prototipagem eletrônica, Figura 45, de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, pinos digitais e analógicos de entrada e saída, entrada USB ou serial, além de possuir um código aberto. Com uma Placa Arduino é possível enviar ou receber informações a partir de qualquer sistema eletrônico conectado a ela. Além disso, seus sensores permitem que você faça a integração com outros dispositivos e também interaja com outros aparelhos. Ou seja, ela pode funcionar por meio da sua própria interface ou com a interação de outros aplicativos instalados no seu computador.

Dentre as vantagens, é possível trabalhar a programação de seus projetos no ambiente online no [creative.arduino.cc/editor](http://creative.arduino.cc/editor) (mais cômodo e indicado) e também é possível instalar e configurar no seu computador para trabalhar Offline, sem necessidade de conexão com a internet.

Figura 45 - Ambiente editor do Arduino



Fonte: Blog Makerhero<sup>43</sup>.

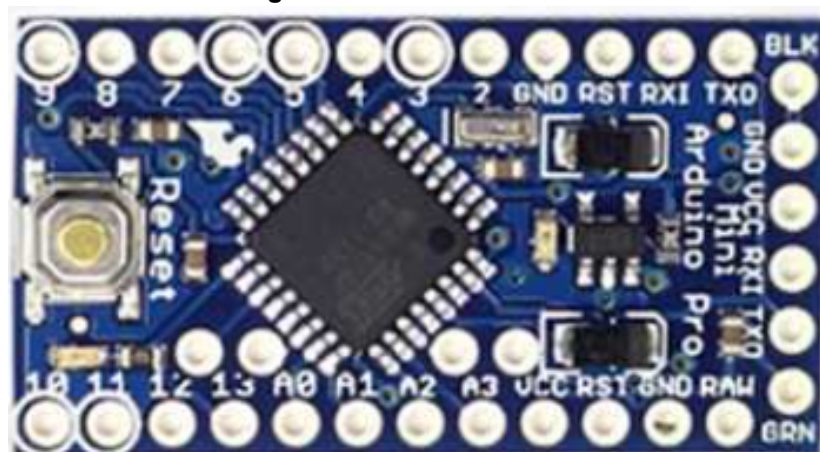
No campo da medicina, da engenharia, da arquitetura e principalmente da automação, têm usado bastante o Arduino. Nós, professores e makers, também utilizaremos essa prototipagem para realizar nossos projetos.

#### 5.7.1 Modelos e tipos de arduino

Atualmente no mercado há diversos modelos e tipos de Arduino para se adaptar a cada estilo de usuário e projetos que desejar construir. Alguns modelos seguem nas Figuras 46 a 51:

1. Arduino Pro Mini;

Figura 46 - Arduino Pro Mini



The Arduino Pro Mini board

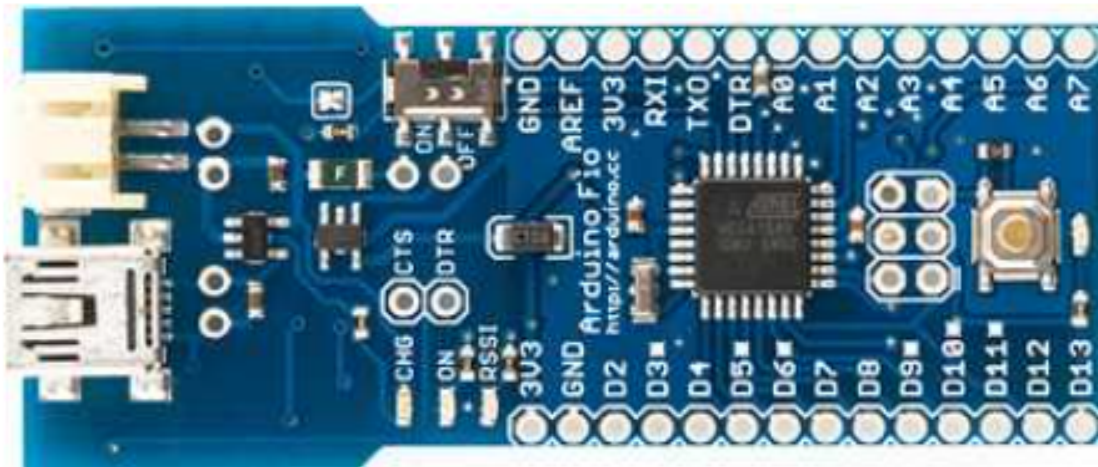
Fonte: Página arduino.cc<sup>44</sup>.

<sup>43</sup> Disponível em: <<https://www.makerhero.com/blog/arduino-web-editor-ide-online/>>. Acesso em 03/2022.

<sup>44</sup> Disponível em: <<https://docs.arduino.cc/retired/boards/arduino-pro-mini>>. Acesso 03/2022.

## 2. Arduino Fio;

Figura 47 - Arduino Fio



A placa Arduino Fio.

Fonte: Página arduino.cc.

## 3. Arduino Micro;

Figura 48 - Arduino Micro

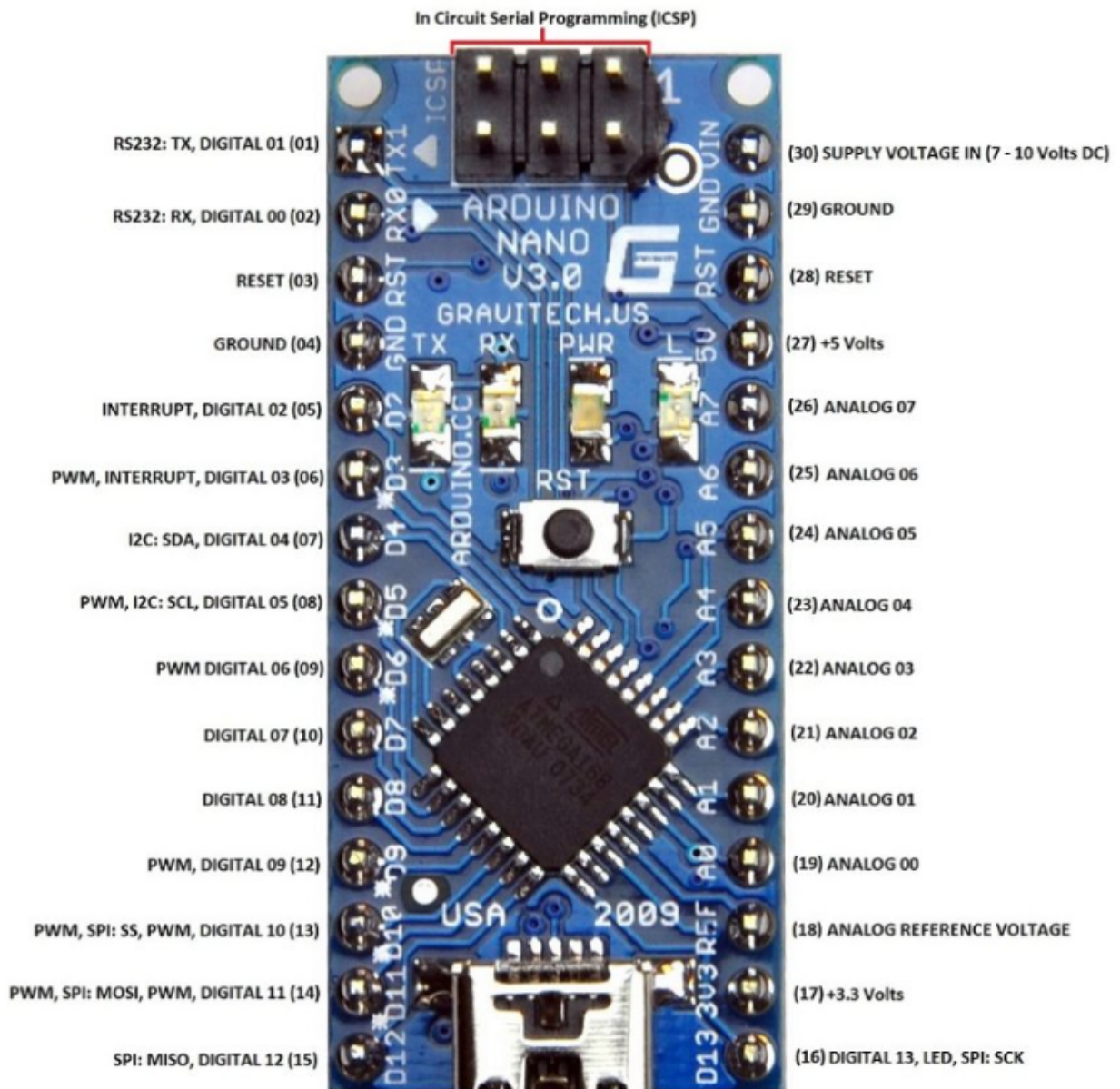


Fonte: Blog Eletrogate<sup>45</sup>.

<sup>45</sup> Blog Eletrogate. Disponível em: <<https://blog.eletrogate.com/gamepad-de-pc-arduino-pro-micro/>>. Acesso em 04/2022.

#### 4. Arduino Nano;

Figura 49 - Arduino Nano



### Arduino Nano V3 - Pin Description

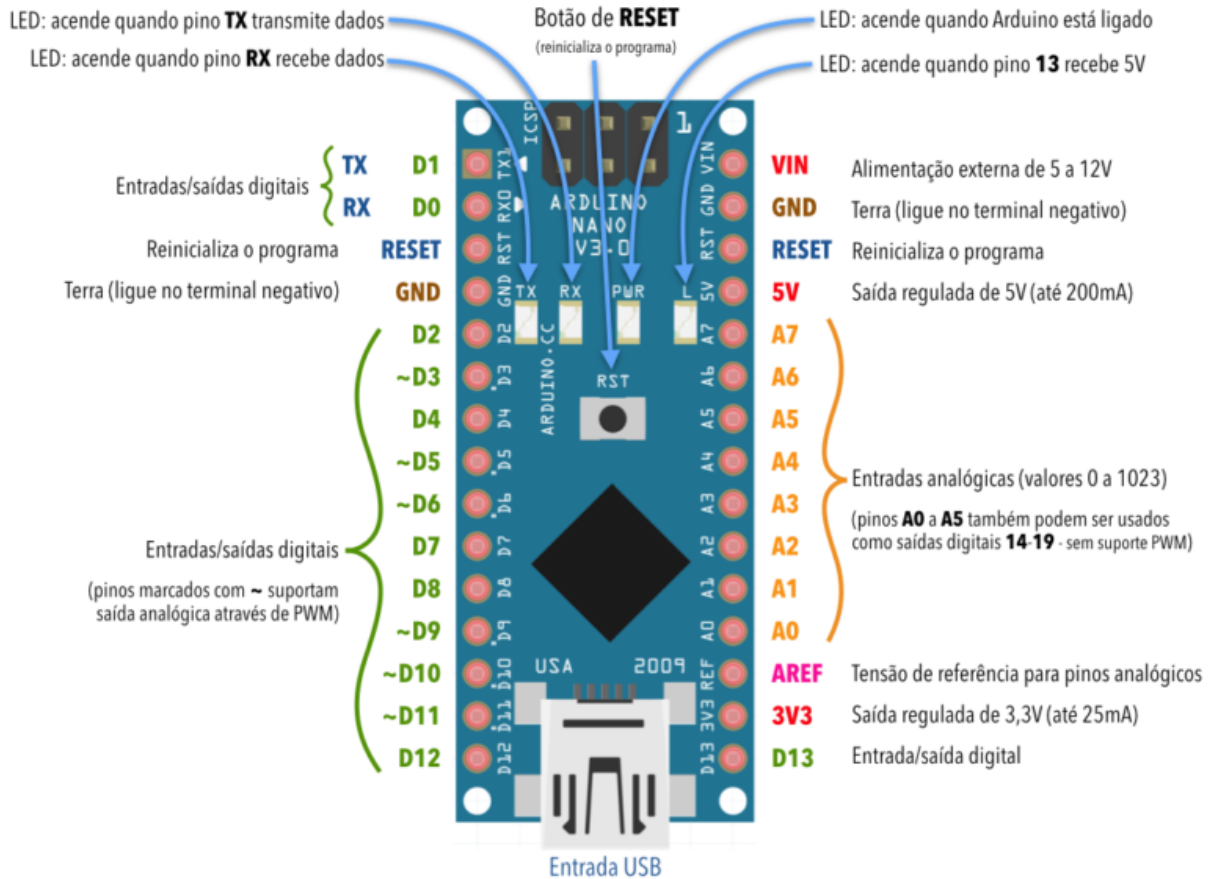
[www.CircuitsToday.com](http://www.CircuitsToday.com)

Fonte: Blog Lobo da Robótica<sup>46</sup>.

<sup>46</sup> Blog Lobo da Robótica. Disponível em: <<https://lobodarobotica.com/blog/arduino-nano-pinout/>>. Acesso em 04/2022.



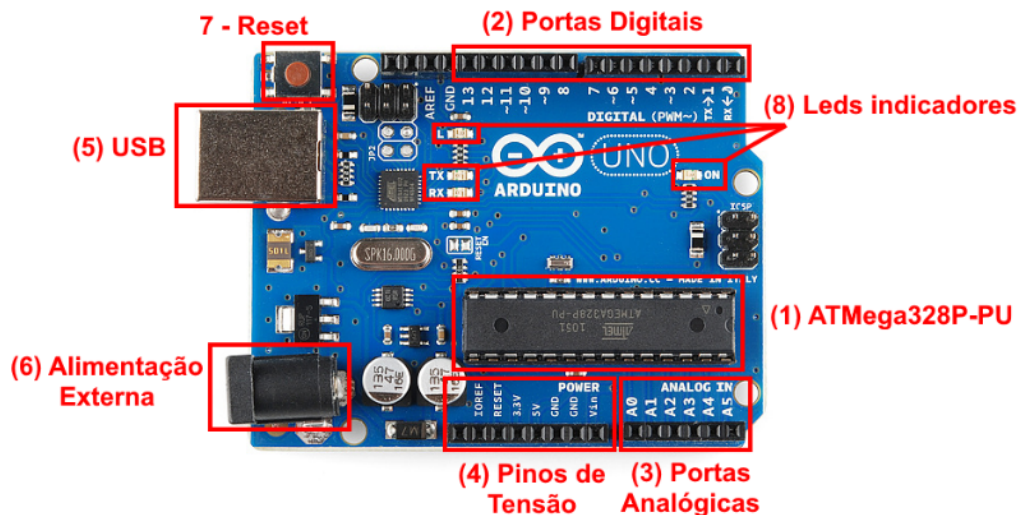
**Figura 50 - Identificação Arduino Nano**



Fonte: Página Eletrônica para artistas<sup>47</sup>.

## 5. Arduino Uno.

**Figura 51 - Identificação Arduino Uno**



Fonte: Página aprendendo física<sup>48</sup>.

<sup>47</sup> Disponível em: <<http://eletronicaparaartistas.com.br/2017/06/page/2/>>. Acesso em 03/2022.

<sup>48</sup> Disponível em: <<http://aprendendofisica.pro.br/pmwiki.php/Main/ArduinoUno>>. Acesso em 03/2022.

## 6. Proposta didática - Circuitos de aprendizagem com oficinas didáticas plugadas e desplugadas

Neste Capítulo apresentaremos uma sequência de seis circuitos de aprendizagem, os quais são compostos por oficinas didáticas que tem por objetivo possibilitar o entendimento e desenvolvimento de habilidades sobre o assunto daquele circuito de aprendizagem.

Mais especificamente, apresentamos oficinas/atividades **desplugadas** (sem uso de recurso digital, quando a atividade não envolve uso de dispositivos digitais) e oficinas/atividades **plugadas** (envolvem uso de recursos digitais, programação, robótica ou ferramentas computacionais). Em algumas dessas atividades pode-se trabalhar sem a necessidade de conexão com a internet no ato da realização das tarefas, enquanto outras dessas atividades pode-se fazer necessária a internet para a instalação do(s) software(s) antes do desenvolvimento dos trabalhos.

Estas oficinas didáticas são diversificadas para valorizar o trabalho do professor, tornando o estudante protagonista de seu conhecimento com uma proposta que possibilita o desenvolvimento da **Competência 5, proposta na BNCC** e assim a possibilidade de desenvolvimento das habilidades do Pensamento Computacional mesmo com o uso de poucos recursos.

Para trabalhar robótica em sala de aula é importante ter materiais disponíveis, mesmo que não seja conforme a lista exigida, mas que tenha materiais necessários para completar cada circuito escolhido a ser executado. Para isso, é importante buscar materiais em lojas especializadas em componentes eletrônicos ou então, adquirir pela Internet ou ainda utilizar materiais reutilizáveis como aqueles que podemos obter desmontando eletrônicos em desuso e que ainda possuem peças em perfeito estado de funcionamento. Na oportunidade, também é apresentado propostas de sites, aplicativos, oficinas didáticas e aplicações/atividades da computação plugada e desplugada dentro e fora de sala de aula.

## 6.1 Computação Plugada

### 6.1.1 Circuito de aprendizagem X Logo

#### 6.1.1.1 Oficina didática - Instalação do X Logo

Nossa primeira oficina didática será com o X logo, uma ferramenta que possibilita trabalho *off line* isto é, sem necessidade de acesso à internet, pois após a instalação do aplicativo (acesso e instalação do X logo pelo link a seguir: <https://projetologo.neocities.org/xlogo.html>) no computador basta abrir o aplicativo instalado e começar a criar. Caso ainda tenha dificuldade para baixar o arquivo acesse o link abaixo, extraia o arquivo zipado e clique no arquivo .exe, após isso é só permitir a instalação conforme abrir cada uma das janelas.

Para conhecermos um pouco sobre o X LOGO, Figura 52, temos nos links abaixo duas apresentações para inspiração:

#### **Para saber mais:**

Acesse: Projetos com LOGO - Apostila originalmente desenvolvida para o ensino fundamental.

[https://docs.google.com/presentation/d/180mPSS3ZQ-Rr\\_O6VSjvAZDewsyjrjMieyJkaeq63XNxl/edit?usp=sharing](https://docs.google.com/presentation/d/180mPSS3ZQ-Rr_O6VSjvAZDewsyjrjMieyJkaeq63XNxl/edit?usp=sharing)

Ambiente LOGO.

<https://docs.google.com/presentation/d/1DTQtrF62hdWG7WC59ITMEQImYCerjlxKEI0mzdk-5jY/edit?usp=sharing>

<https://drive.google.com/file/d/1IbzhA-6G2fxl-VGnZC1TdGapphuwLwVO/view>

<https://www.if.ufrj.br/~carlos/infoenci/logo.html>

Figura 52 - Ambiente X LOGO



Fonte: Projeto Logo<sup>49</sup>

Concluída a instalação do aplicativo podemos iniciar.

Movimentando a tartaruga.

A tartaruga é geralmente um pequeno desenho que pode se deslocar pela tela do computador desenhando e colorindo objetos, seus movimentos são controlados por linhas simples de comandos de programação.

Para ter uma ideia dos comandos Logo que movimenta a tartaruga, digite na caixa de entrada.

```
parafrente 50
```

Com isto a tartaruga deve andar 50 passos (pontos da tela) para a frente.

Escreva agora.

```
paraesquerda 90
```

<sup>49</sup> Disponível em: <<https://projetologo.neocities.org/xlogo.html>>. Acesso em 03/2022.

Note que a tartaruga girou 90 graus para a esquerda.

Tente também.

```
paratrás 80
paradireita 270
```

Seu código completo ficou:

```
parafrente 50
paraesquerda 90
paratrás 80
paradireita 270
```

Estes são os comandos básicos para mover a tartaruga. Como muitos outros comandos Logo, eles podem ser escritos de forma abreviada:

```
parafrente = pf
paratrás = pt
paraesquerda = pe
paradireita = pd
```

Dessa forma o código fica:

```
pf 50
pe 90
pt 80
pd 270
```

A tartaruga deixa um rastro por onde passa, como se estivesse usando um lápis. Se queremos movê-la sem desenhar, empregamos o comando “usenada”.

Verifique o que acontece com,

```
usenada
pf 50
```

A tartaruga deve ter andado sem deixar rastros. Se quisermos voltar a desenhar sobre a tela usamos o comando “uselápis”. Por exemplo:

```
uselápis
pt 70
```

A tartaruga pode ficar invisível. Para isto basta executar o comando

```
desapareçatat
```

Para torná-la visível execute

```
apareçatat
```

Para limpar a tela, apagando todos os desenhos, e recolocar a tartaruga na sua posição inicial, use o comando

```
tartaruga
```

Que pode ser abreviado para `tat`. Ou simplesmente aperte o botão "TAT" que está ao lado da caixa de saída.

### 6.1.1.2 Oficina didática - Desenhar um quadrado

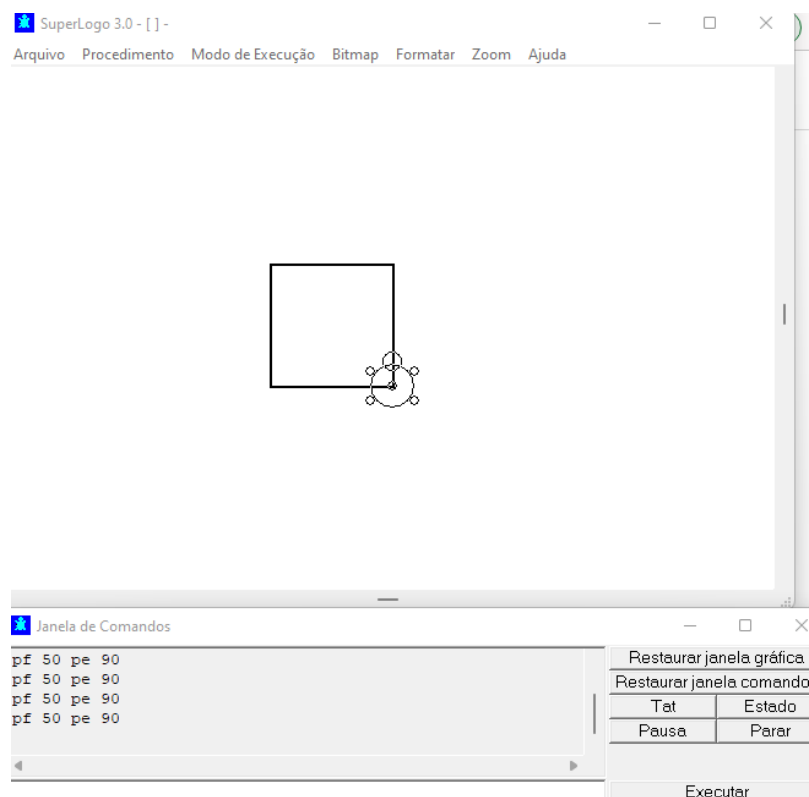
Podemos fazer a tartaruga desenhar um quadrado, Figura 53, com as instruções:

```
pf 50 pe 90  
pf 50 pe 90  
pf 50 pe 90  
pf 50 pe 90
```

(note como é possível escrever mais de um comando Logo por linha).

Entretanto, é bem mais simples usar o comando `repita`, `repita 4 [pf 50 pe 90]`.

**Figura 53 - Desenhando um quadrado com X LOGO**

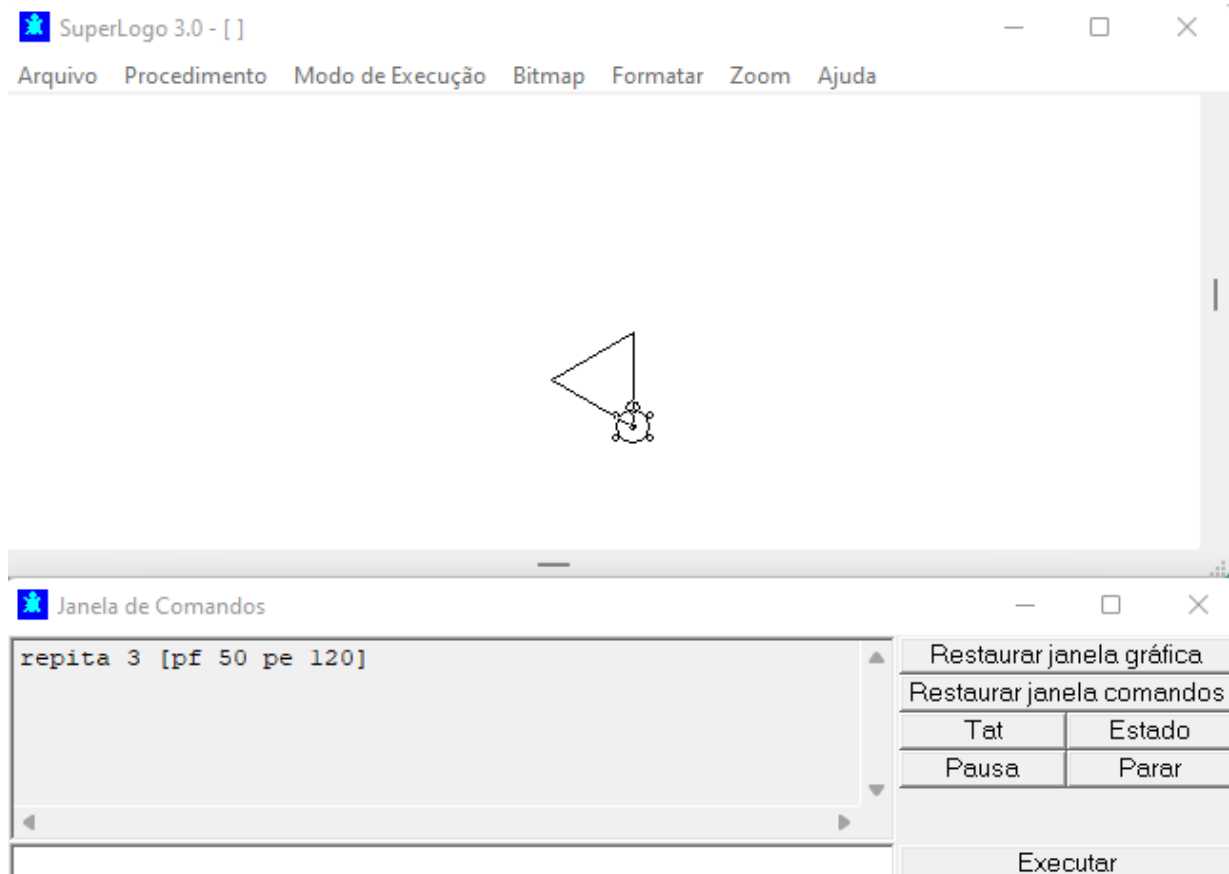


Fonte: Aatoria própria.

### 6.1.1.3 Oficina didática - Desenhar um triângulo

Um triângulo, Figura 54, pode ser traçado com `repita 3 [pf 50 pe 120]`.

**Figura 54 - Desenhando um triângulo com X LOGO**

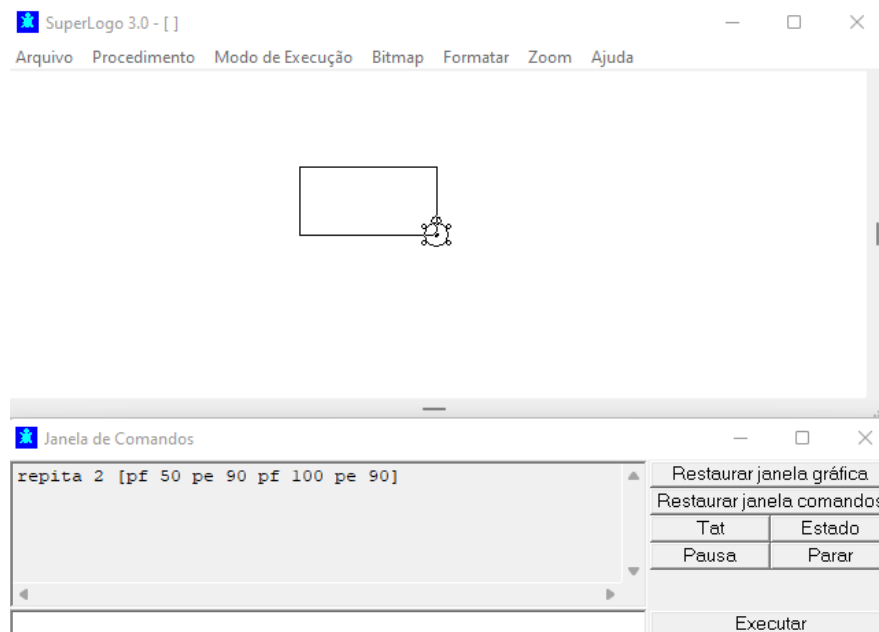


Fonte: Autoria própria.

#### 6.1.1.4 Oficina didática - Desenhar um retângulo

Um retângulo, Figura 55, é obtido via `repita 2 [pf 50 pe 90 pf 100 pe 90]`

**Figura 55 - Desenhando um retângulo com X LOGO**



Fonte: Autoria própria.

O comando “contevezes” (ou `cv`) fornece o número de repetições já realizadas pelo `repita`.

Tente, por exemplo, para análise dos resultados:

```
repita 3 [escreva contevezes]
```

Aqui teremos como retorno, a sequência 1, 2, 3 na janela de comandos.

Ou,

```
repita 720 [pf 10 pe contevezes]
```

ou ainda

```
repita 200 [pf contevezes pe 90]
```



Aqui teremos um desenho com 200 quadrados em espiral, visualmente na janela gráfica.

```
E, repita 1800 [pf 10 pe contevezes + .1]
```

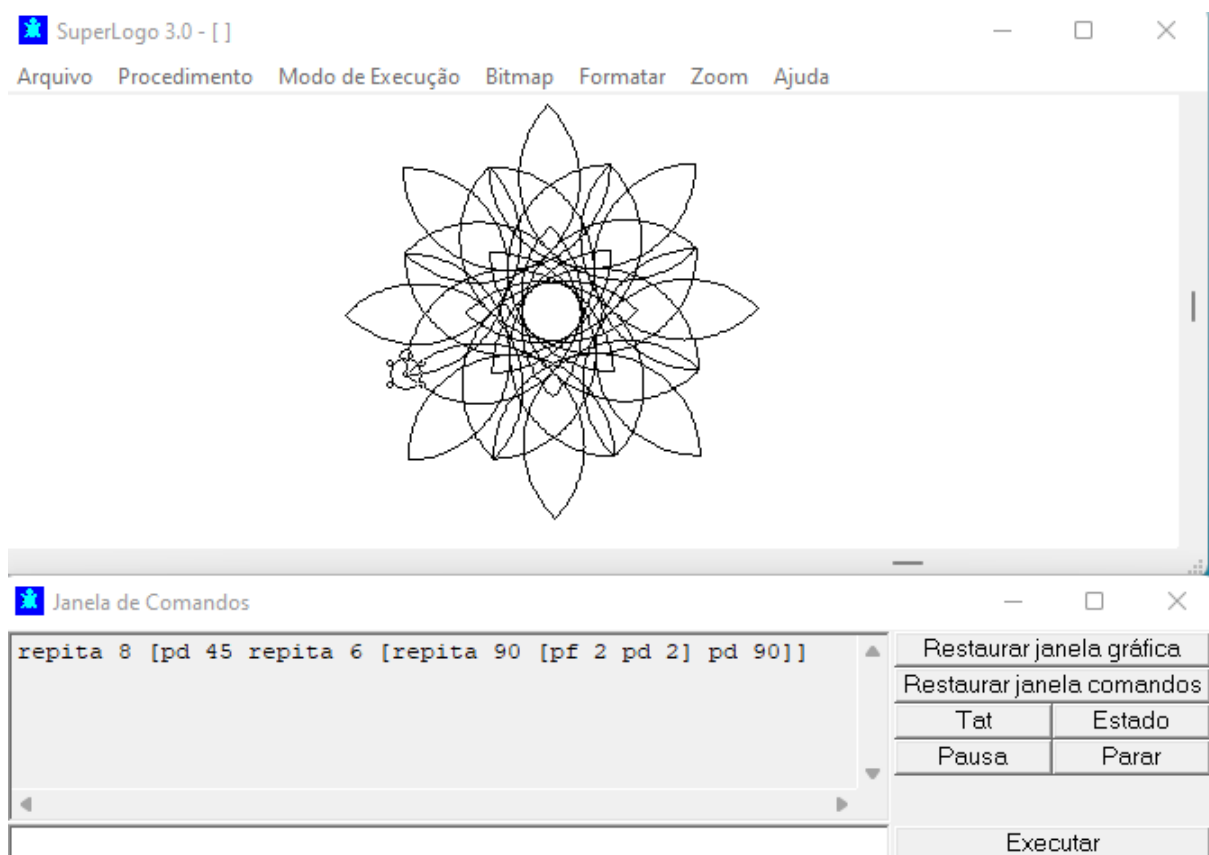
Dentro da lista de comandos do `repita`, Figura 56, podem estar outros `repita`. Veja o que acontece com,

```
repita 6 [pe 60 repita 4 [pf 50 pe 90]]
```

E

```
repita 8[pd45 repita 6[repita 90[pf2 pd2]pd90]]
```

**Figura 56 - Desenhando com X LOGO usando comando `repita`**



Fonte: Autoria própria, adaptado de Aguiar<sup>50</sup>.

<sup>50</sup> Aguiar, Carlos Eduardo. Informática no ensino da física. Disponível em: <<https://www.if.ufrj.br/~carlos/infoenci/notasdeaula/logo-intro/logointr.html>>. Acesso em 05/2022.

#### 6.1.1.5 Oficina didática - Desenhar o Círculo usando procedimentos

Uma das características mais importantes do Logo é a facilidade com que podemos estender a linguagem, criando novos comandos e operações (ou procedimentos). Por exemplo, vamos definir um comando chamado quadrado que desenha um quadrado na tela.

Antes, vamos verificar se já não há um procedimento com este nome. Escrevendo,

```
quadrado
```

Na caixa de entrada obtemos como resposta a mensagem. AINDA NÃO APRENDI QUADRADO.

Ou seja, a tartaruga (ou melhor, o Logo) não conhece tal procedimento. Para definir o comando, clique em PROCEDIMENTO / EDITAR na barra de menu. Na janela que se abrir escreva o nome do procedimento que deseja definir - no caso, quadrado - e aperte o botão OK.

A janela do editor de procedimentos aparecerá então, com o início e o fim da definição do procedimento quadrado já escritos:

```
aprenda quadrado
fim
```

A definição propriamente dita do procedimento deve ser inserida entre o aprenda e o fim. No nosso caso teríamos,

```
aprenda quadrado
repita 4 [pf 50 pe 90]
fim
```

Escolha em seguida o ítem ÁREA DE TRABALHO / ATUALIZAR na barra de menu do editor e salve a definição do comando na memória do computador. Clicando em ÁREA DE TRABALHO / SAIR você fecha o editor de procedimentos.

Escreva agora

```
quadrado
```

Na janela de comandos e mande executar. Ao invés de enviar uma mensagem de erro como antes, a tartaruga agora desenhará um quadrado de lado 50.

Esta definição de quadrado não é muito útil, pois obtemos sempre um quadrado do mesmo tamanho (50 pontos). Não podemos desenhar um quadrado de lado 100, ou 10. Seria melhor que o quadrado fosse semelhante, por exemplo, ao comando `parafrente`, que tem um parâmetro que determina o tamanho do deslocamento. Assim, quadrado 100 faria um quadrado de lado 100. É fácil criar esta versão aperfeiçoada do comando. Para isto, volte ao PROCEDIMENTO / EDITAR da barra de menu, e escolha o procedimento `quadrado` na janela de opções.

O procedimento definido anteriormente vai aparecer na janela do editor. Mude-o para,

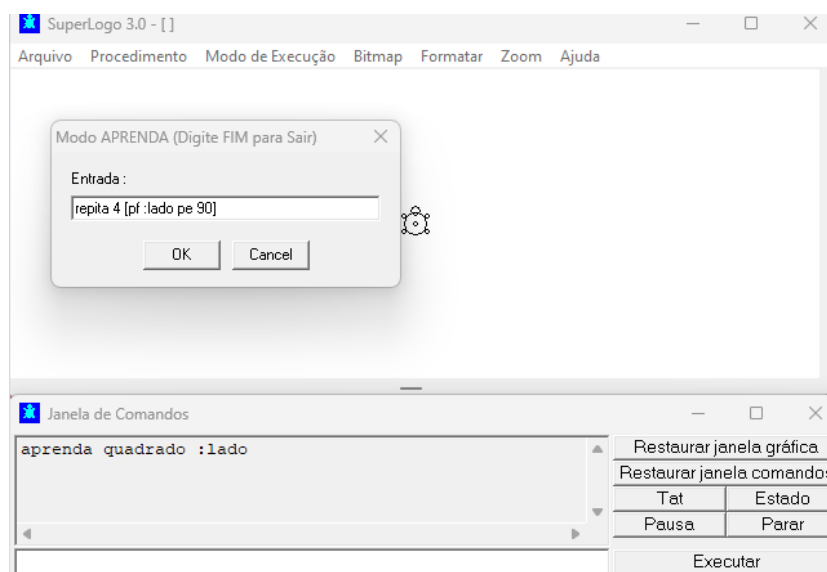
```
aprenda quadrado :lado
repita 4 [pf :lado pe 90]
fim
```

e salve o resultado.

```
quadrado 100 ou quadrado 1
```

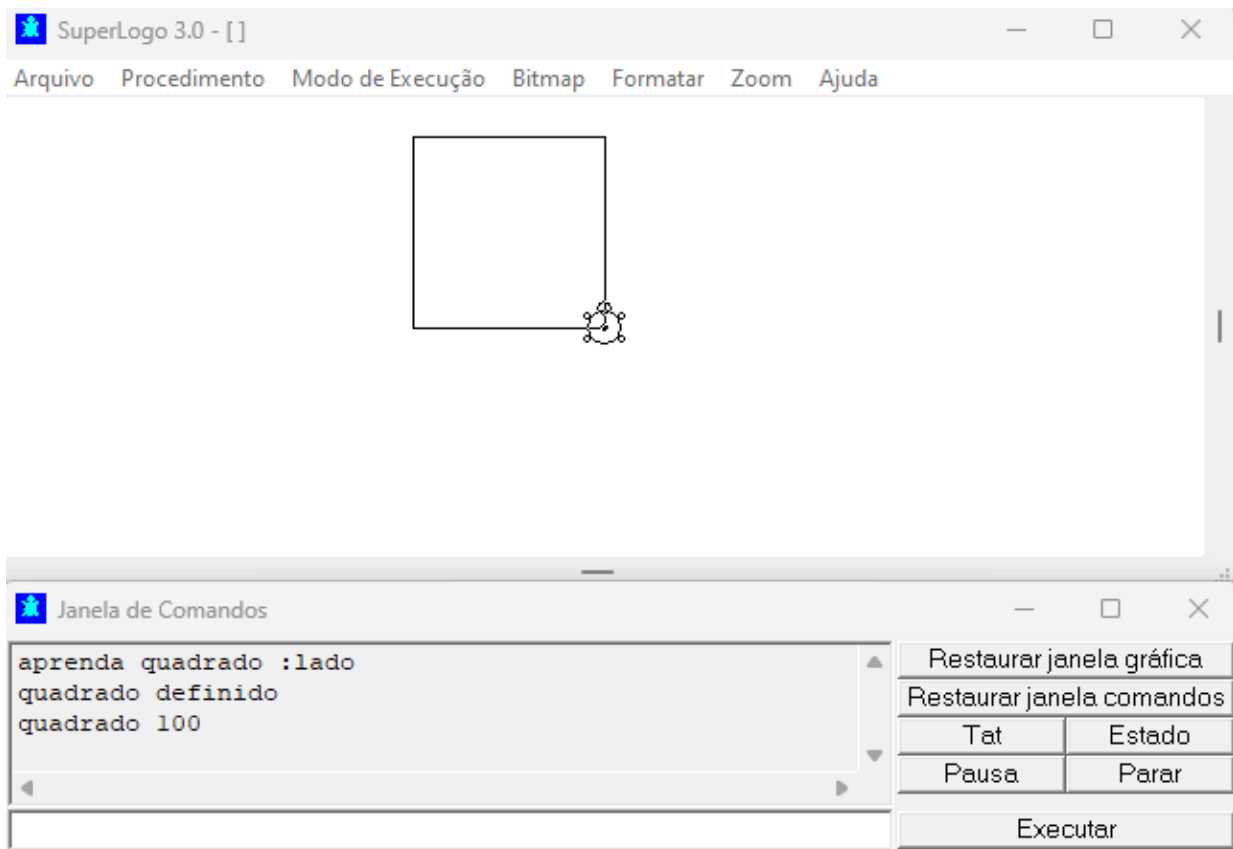
Para facilitar, veja as Figuras 57 e 58 com os resultados desses comandos.

**Figura 57 - Comando aprenda e repita**



Fonte: Autoria própria.

Figura 58 - Comando quadrado 100



Fonte: A autoria própria.

Agora, se tentarmos, obteremos quadrados de lados 100 e 10 respectivamente. Veja também o que acontece com a linha de comando a seguir. Resultado do comando na Figura 59.

```
repita 50 [quadrado 2*contevezes]
```

**Figura 59 - Comando contevezes**



Fonte: Autoria própria.

Resolução do desafio: Procedimento que desenha um círculo com raio escolhido pelo usuário:

```
aprenda círculo :r
  repita 360 [pf pi * :r / 180 pd 1]
fim
```

```
aprenda círculo2 :r
  elipse :r
  :r
fim
```

**Para saber mais:**

Acesse outras atividades com X-logo em:

<http://www.gestaoescolar.diaadia.pr.gov.br/arquivos/File/tutoriais/xlogo.pdf>

[http://www1.pucminas.br/imagedb/documento/DOC\\_DSC\\_NOME\\_ARQUI20140528143213.pdf](http://www1.pucminas.br/imagedb/documento/DOC_DSC_NOME_ARQUI20140528143213.pdf)

Fonte dessa oficina didática: <<https://www.if.ufrj.br/~carlos/infoenci/notasdeaula/logo-intro/logintr.html>>. Acesso em 07/2023.

## 6.1.2 Circuito de aprendizagem - Code.org

### 6.1.2.1 Oficina didática - AngryBirds

Nesta atividade é possível aprender a programar de uma forma divertida.

Materiais:

- Computador ou tablet ou ainda um celular;
- Acesso a internet.

No ambiente Code.org<sup>51</sup>, Figura 60, crie uma conta pessoal para que seu progresso seja registrado no ambiente online. Feito isso, basta seguir os passos, de acordo com cada uma das lições sequenciais do programa de lições.



Fonte: Autoria própria por code.org.

Perceba que as lições iniciais são bastante intuitivas, basta arrastar os blocos e assim que estiver pronto a sequência de comandos, clicar em executar e é analisado se a solução irá solucionar o problema ou se ainda existe alguma observação a ser reparada.

<sup>51</sup> Code.org. Disponível em: <<https://studio.code.org/hoc/1>>. Acesso em 03/2022.

Caso precise reiniciar a sua sequência de comandos, clique em começar do início. Após isso, basta reiniciar e executar novamente até que abra a lição seguinte.

Para que você possa conhecer um pouco mais sobre os códigos de programação, Figura 61, em inglês, que podem substituir os blocos, basta clicar em “Mostrar código”.



Fonte: Autoria própria por code.org.

### **Para saber mais sobre o ANGRY BIRDS:**

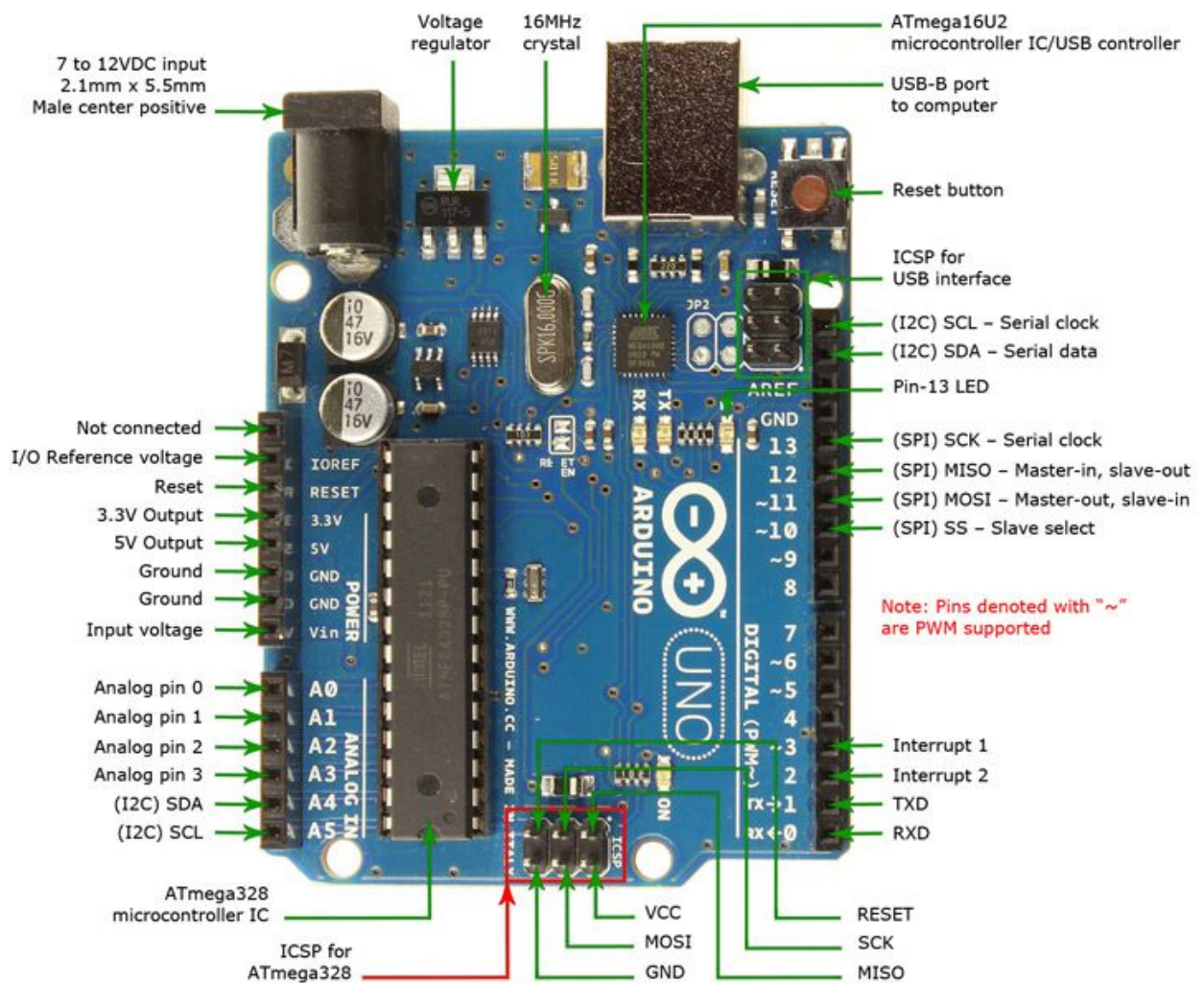
Para quem ainda possui dificuldades, pode acompanhar as videoaulas em:  
<https://www.youtube.com/watch?v=B3LEAf0Tz08&list=PLQ-qrOvON8ZMc9Qodh-WDWWeKoz-TchI2&index=1>

### 6.1.3 Circuito de aprendizagem Tinkercad um simulador Arduino

#### 6.1.3.1 Oficina didática - Tinkercad e Arduino - Primeiros passos

Vejamos algumas atividades que utilizam a placa Arduino UNO, Figura 62, com o simulador Tinkercad.

**Figura 62 - Entradas e saídas do Arduino Uno**



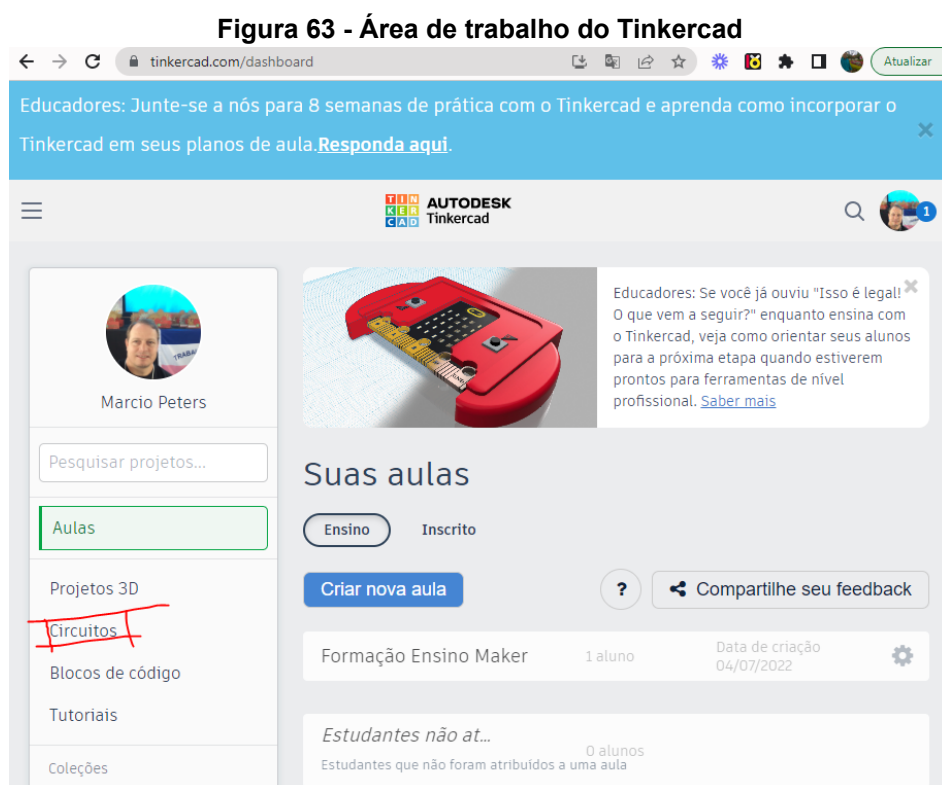
Fonte: Página um.pro<sup>52</sup>

<sup>52</sup> Página um.pro, disponível em: <<http://www.um.pro.br/arduino/index.php?c=ArduinoUno>>. Acesso em 12/2021.



Agora é o momento de conhecer a plataforma Tinkercad<sup>53</sup>, Figura 63. Para iniciar faça login na plataforma (caso não possua conta no ambiente, faça o cadastro gratuitamente utilizando seu e-mail pessoal), escolha um dos perfis, conforme desejar, “Educador”, “Estudante” ou “Conta pessoal” e habilite seu acesso.

Agora, já logado no ambiente, você poderá editar seu perfil, incluir foto e outras informações básicas.



Fonte Autoria própria.

Na opção “Circuitos” é possível conhecer projetos de outros autores e também iniciar o seu próprio projeto. Vamos lá?

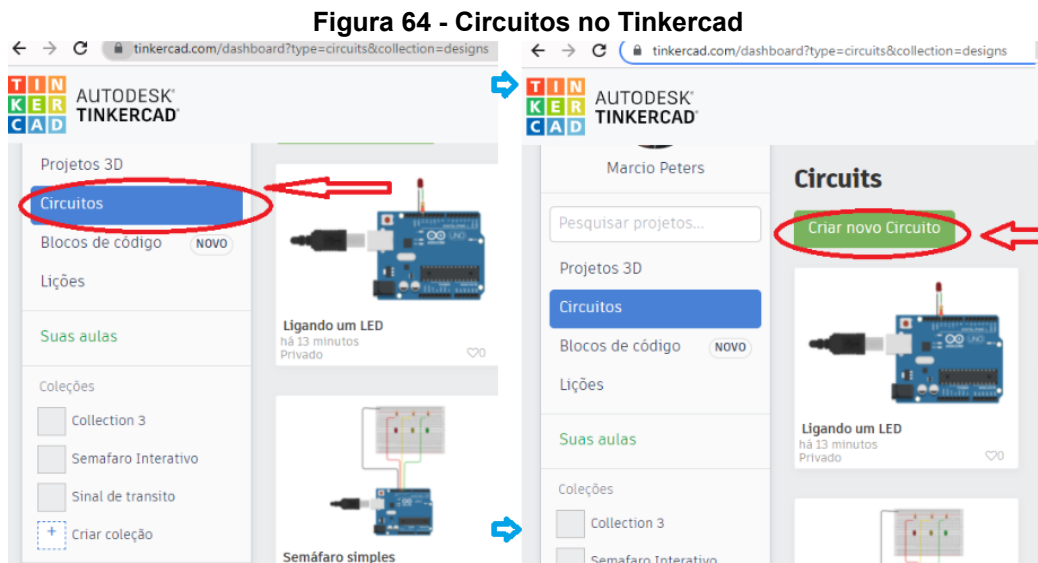
<sup>53</sup> Tinkercad, disponível em: <<https://www.tinkercad.com/dashboard>>. Acesso em 05/2022.

### 6.1.3.2 Oficina didática - Acender um LED

Nesta oficina vamos usar o Tinkercad e acender um LED.

A primeira tarefa é: Abra a plataforma Tinkercad, Figura 64, (<https://www.tinkercad.com/dashboard>) e faça login com a conta já criada na oficina, acender um LED.

Já logado escolha circuitos, conforme imagem a seguir:



Fonte: A autoria própria - Tinkercad 2022.

Em seguida, escolha “Criar novo circuito”.

Agora no seu projeto, escolha “Componentes Básico” depois “Arduino”, Figura 65, segue abaixo:



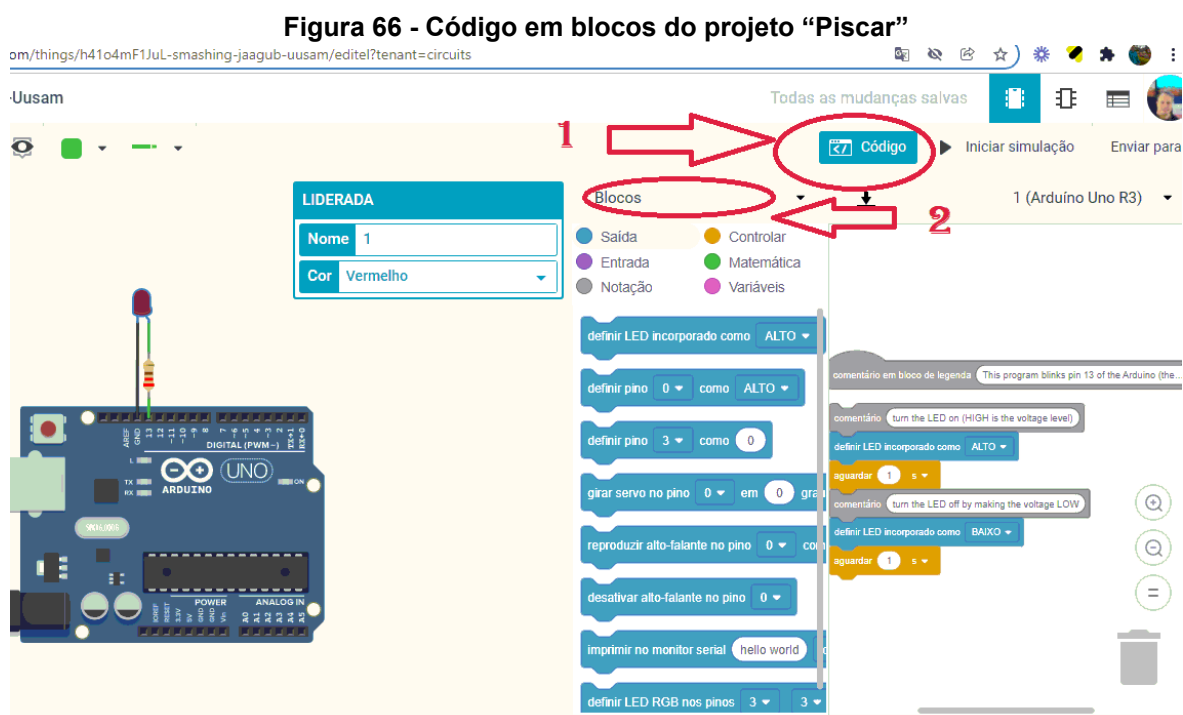
Fonte: A autoria própria - Tinkercad 2022.

Agora, selecione com clique e arraste para a janela do seu projeto a opção “Piscar”, conforme Figura 65.

Neste momento, seu projeto já está pronto, mas vamos fazer alguns testes para que possamos entender melhor.

Conforme imagem abaixo, escolha “Código” => “Blocos”.

Veja que a linguagem de programação que está sendo utilizada é a linguagem em blocos, Figura 66, que é uma linguagem simples e facilita muito para os iniciantes em programação.

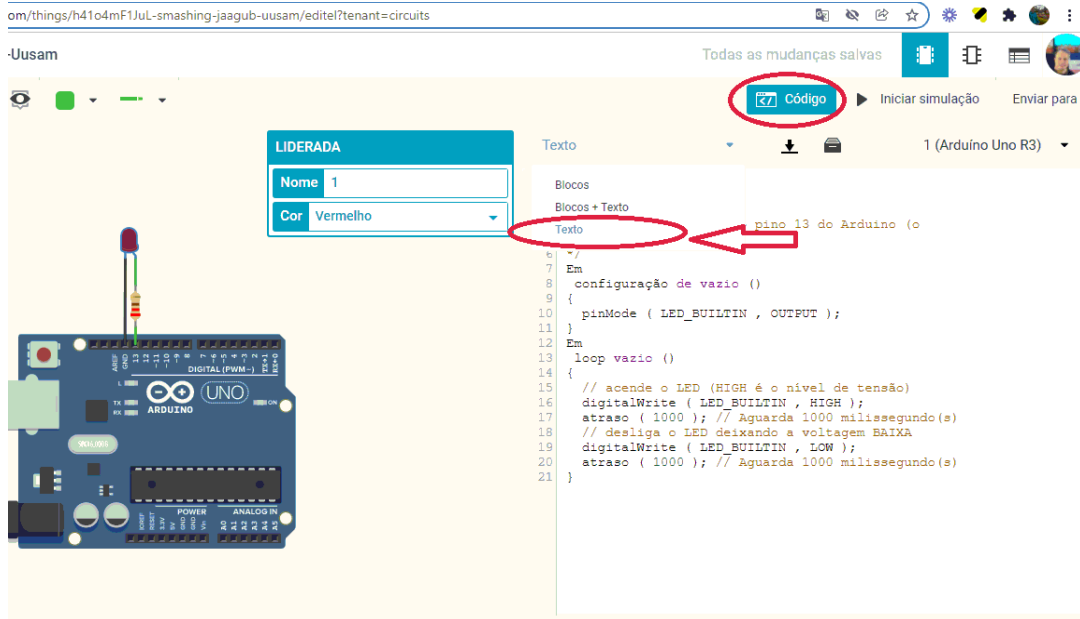


Fonte: Autoria própria - Tinkercad 2022

No entanto, vamos explorar um pouco mais escolhendo “Código” => “Texto”.

Veja que o mesmo projeto está escrito na linguagem C, Figura 67, com as mesmas características e funções.

Figura 67 - Código em texto do projeto “Piscar”

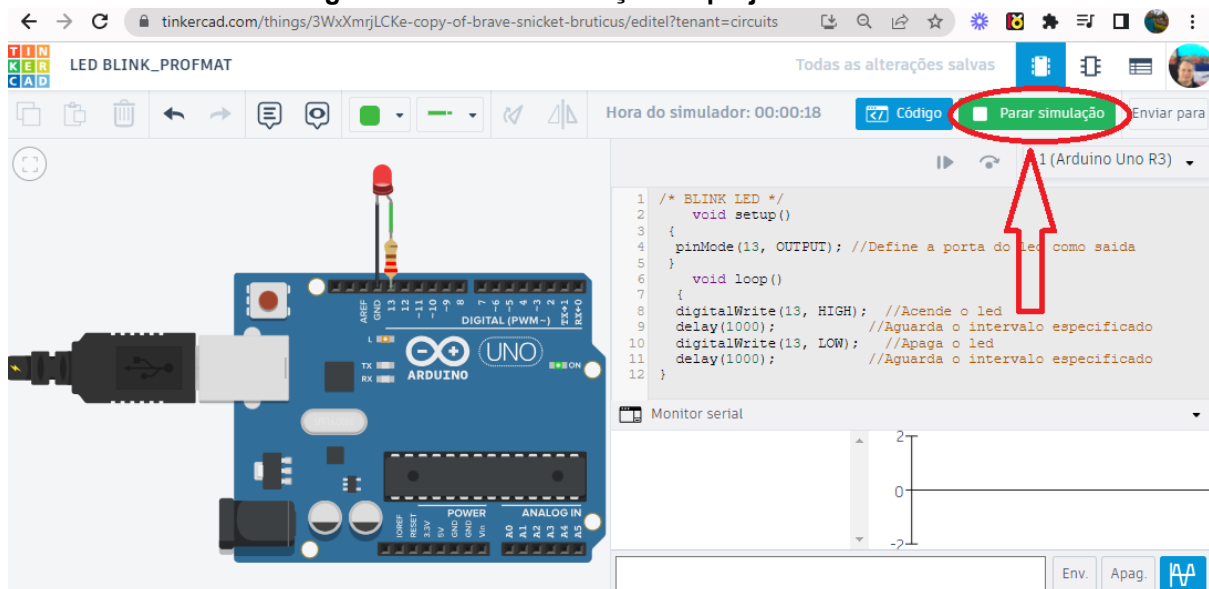


Fonte: Autoria própria - Tinkercad 2022.

Assim, você pode escolher qual tipo de linguagem utilizar conforme desejar.

Agora vamos para a parte da simulação: clique em “Iniciar simulação”, Figura 68, e observe que o LED irá piscar com delay (tempo) definido na programação.

Figura 68 - Iniciar simulação do projeto “Piscar”



Fonte: Autoria própria - Tinkercad 2022.

Agora, altere o tempo (delay) conforme desejar e veja o que acontece após iniciar a simulação novamente.

Código da programação<sup>54</sup>, veja abaixo:

```
/* BLINK LED */
  void setup()
  {
    pinMode(13, OUTPUT); //Define a porta do led como saída
  }

  void loop()
  {
    digitalWrite(13, HIGH); //Acende o led
    delay(1000);           //Aguarda o intervalo especificado
    digitalWrite(13, LOW); //Apaga o led
    delay(1000);           //Aguarda o intervalo especificado
  }
}
```

#### 6.1.4 Circuito de aprendizagem Arduino, o ambiente Editor de Códigos

##### 6.1.4.1 Oficina didática - Instalação do Arduino

Esta é uma oficina cujo objetivo é facilitar o ensino de robótica a ponto de poder ser executada inclusive por pessoas que não sabem programar, aprendizes, desenvolvedores e curiosos em geral.

A inteligência de toda a brincadeira está na placa eletrônica chamada Arduino que irá distribuir todos os comandos, ou instruções, que inventarmos em nossas aventuras pela robótica. É o cérebro do nosso futuro robô. Existem opções de placa robótica, mas aqui vamos usar esse Arduino por ser uma das mais acessíveis e simples.

Para iniciar, vamos testar nossa placa.

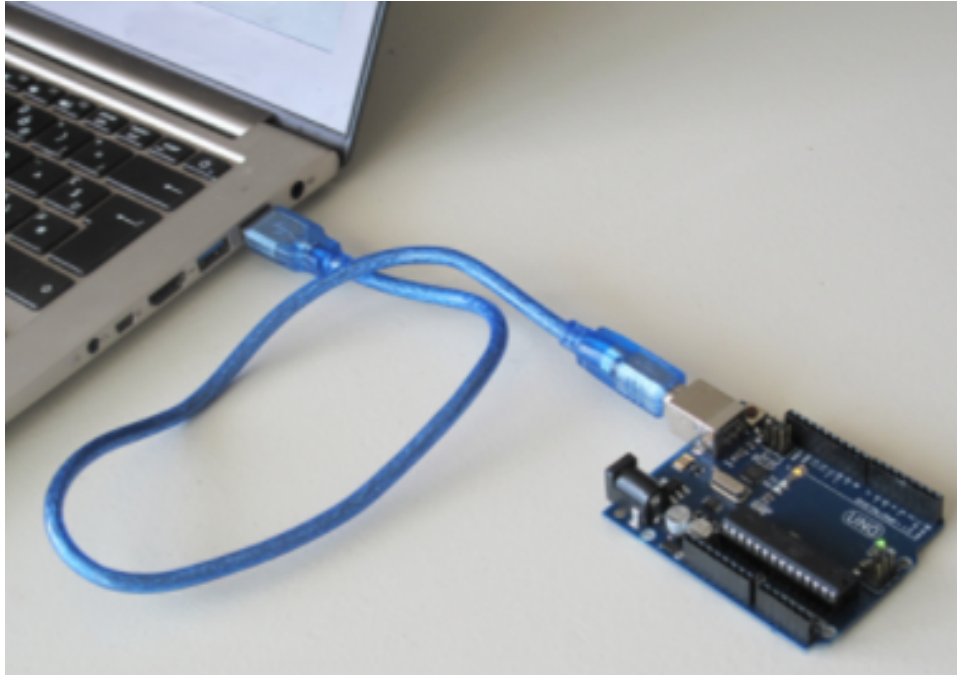
Use o cabo USB com a placa Arduino para ligar a energia do computador.

---

<sup>54</sup> O código da programação irá aparecer configurado com o fundo escuro propositalmente, para indicar ao leitor que esta parte do texto foi o código utilizado na parte prática.

Assim que ligar o cabo da placa na saída USB do computador, Figura 69, que libera energia, duas luzes de sua placa Arduino acenderão automaticamente. Feito isso, então ok! Está tudo bem com sua placa Arduino até agora.

**Figura 69 - Placa Arduino conectada ao computador**



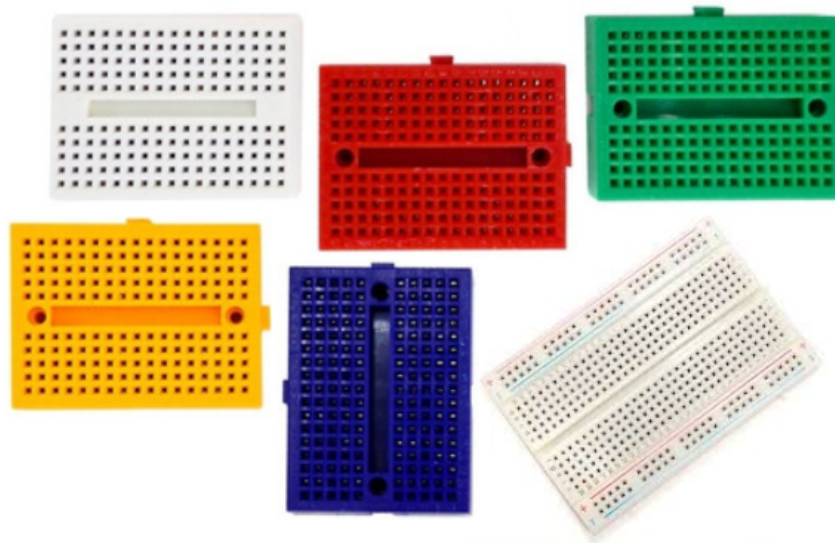
Fonte: Página medium.com<sup>55</sup>

Prototipar é uma técnica que consiste em desenvolver o protótipo de alguma ideia ou invenção, é uma forma de visualizar a sua ideia antes de ser concretizada, isto é, antes de pôr em prática é importante testar de uma forma mais prática todas as etapas de execução. Antes de montarmos um robô de verdade, com cabeça, braços, pernas, etc., é importante prototiparmos esse robô que irá executar as ações desejadas. Para isso, na robótica usamos a Protoboard, Figura 70, que é uma placa bem simples, sem inteligência, que só contém furinhos interligados entre si em sequências verticais e outras horizontais. É nessa placa que vamos fixar os componentes que desejarmos prototipar. Dentro de cada sequência temos que os 5 furos estão localizados internamente de maneira a formar uma única conexão elétrica.

---

<sup>55</sup> Disponível em: <<https://medium.com/como-programar-em-1-dia/como-aprender-rob%C3%B3tica-em-1-dia-1a3edd04db7e>>. Acesso em 05/2022.

**Figura 70 - Placas protoboard**



Fonte: Autoria própria

### **Instalar no computador o editor de códigos Arduino.**

Muito além de uma placa inteligente e educativa, o Arduino possui todo um portal online para poder brincar de forma simples e divertida. É possível instalar no computador esse editor de código e posteriormente trabalhar sem a necessidade de acesso à internet. Acesse a plataforma oficial do Arduino em: <https://arduino.cc>

### **Instalando o driver da placa.**

Para que o seu computador se comunique com uma placa Arduino e para que você possa enviar para ela os programas que deseja executar, é preciso instalar o plugin do Arduino em seu computador.

Dentro do portal do Arduino, vá ao menu “Software > Online Tools”, e depois selecione a opção “Getting Started”. Abrindo a tela selecione a opção “Autodetect Arduino Board”.

Conecte-se, após clicar em “Getting Started”, selecione o botão “LOGIN” (ou o botão “START” se já estiver logado).

Faça seu login na plataforma, criando uma conta (caso não possuir ainda) ou clicando no botão de logar com o Google.

Na tela, insira um nome do usuário desejado, selecione o próximo tipo de privacidade que lhe interessar e clique em “Create Account”.

Depois de logado, faça o download do arquivo do plugin clicando no botão “Download”.

Execute então o arquivo do plugin baixado em seu computador, dando um duplo clique no arquivo.

### **Programa de instalação do plugin do Arduino.**

Ao abrir o arquivo de instalação do plugin do Arduino, a janela de instalação do plugin se abrirá. Clique em "Avançar", aceite a licença na próxima tela, aceite o diretório de instalação do Chrome na próxima, e clique em "Sim" se estiver usando um navegador diferente do Safari. Clique em "Avançar" novamente e deixe instalar o plugin.

Caso acione o sistema operacional Windows.

E depois também, permita o acesso de segurança para que o Windows instale o plugin.

Finalize a instalação se tudo correr bem.

### **Reconhecimento do plugin.**

Após finalizada a instalação do plugin, volte ao navegador, na tela que estava fazendo a instalação e clique no botão "Try Again" , para refazer a verificação da instalação.

O instalador irá reconhecer automaticamente o novo plugin do Arduino instalado em seu computador.

Você inclusive poderá visualizar o plugin instalado na barra de tarefas de seu sistema operacional.

### **O editor de código.**

O portal online do Arduino também nos oferece uma excelente ferramenta para criar os programas que enviaremos para a placa, para que ela execute os comandos que desejarmos.

No portal online, [arduino.cc](http://arduino.cc) , vá ao menu "Software > Online Tools" e dessa vez selecione a opção "Arduino Web Editor". O editor de código do Arduino aparecerá.

Observe que a placa Arduino modelo UNO já foi elaborada pelo plugin dentro do Editor.

Observe as funções (primeiras funções) de código `setup()` e `loop()` que já foram criadas na primeira tela do editor de código do Arduino. Elas são as principais funções de comunicação com a placa. Sendo `setup()` a função para as configurações iniciais de nosso programa, e `loop()` a função que fica rodando enquanto a placa estiver conectada.



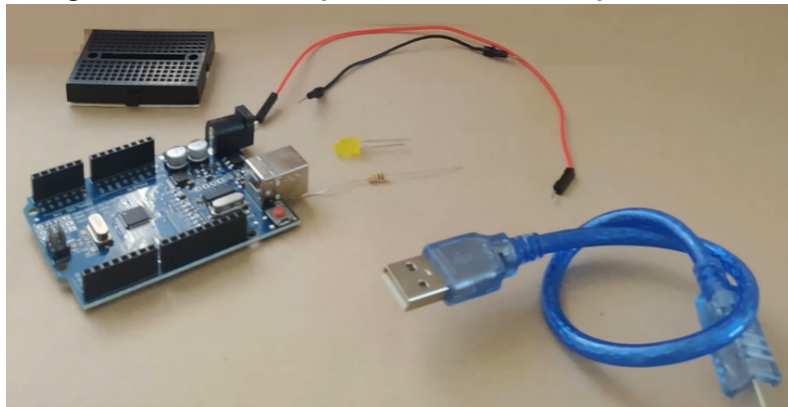
#### 6.1.4.2 Oficina didática - Programando LED Blink

Para esta oficina você vai precisar do seguinte kit de componentes eletrônicos para Arduino, Figura 71.

Material:

1. uma placa Arduino Uno R3 com cabo USB;
2. uma protoboard;
3. um LED;
4. um resistor de 150 ohms;
5. dois Jumpers macho/macho;
6. um notebook/desktop para cada aluno ou grupo de alunos.

**Figura 71 - Kit de componentes eletrônicos para Arduino**



Fonte: Aatoria própria.

Após conferir o material necessário, podemos então começar a trabalhar!

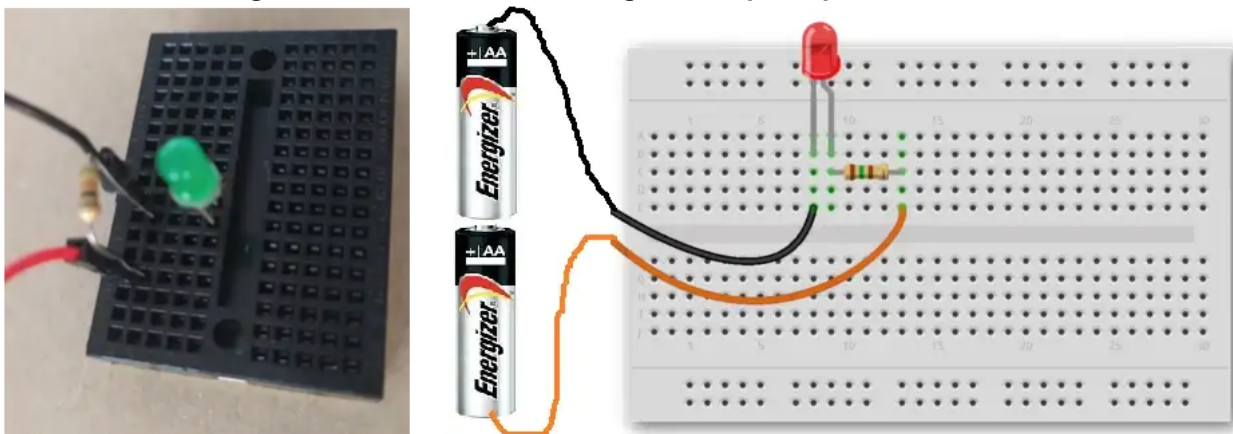
Agora que já conhecemos um pouco mais, vamos iniciar a montagem do nosso primeiro circuito elétrico com uma protoboard e acender uma luz de LED usando um resistor, os fios de eletricidade (jumpers) e duas pilhas comuns (AA, 1,5 volts cada).

1. Conecte uma lâmpada com uma ponta em cada série de 5 furos;
2. Conecte uma ponta de um dos jumpers na mesma série de uma das pontas da lâmpada;
3. Agora conecte uma ponta do resistor na outra série da ponta da lâmpada;
4. Depois conecte uma ponta do outro jumper na mesma série da segunda ponta do resistor;

5. Por último, una as pontas das pilhas AA, observando as polaridades opostas e depois una as pontas extras dos jumpers às pontas das pilhas, sempre observando as polaridades opostas (+ e -). Caso não acenda, troque a polaridade invertendo os fios nas pontas das pilhas, Figura 72.

Observação: **NÃO** ligue a lâmpada diretamente na energia sem o resistor, pois pode acontecer a perda da lâmpada com voltagem menor do que as pilhas e irá queimá-la. O resistor reduz a energia, e funciona como um controlador mediando entre ambos.

Figura 72 - Circuitos elétricos ligados na placa protoboard



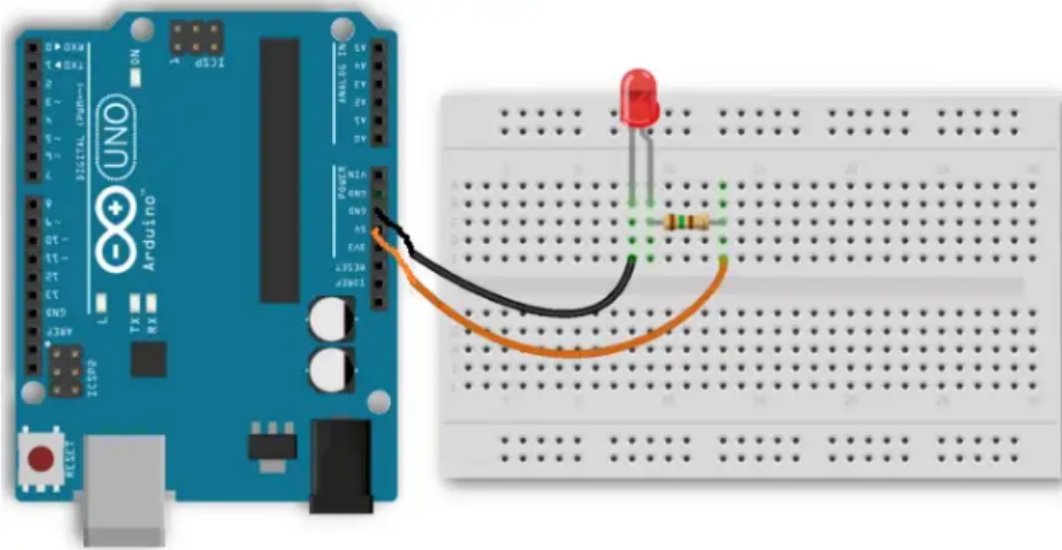
Fonte: Página medium.com<sup>56</sup>.

Outra opção para que esse circuito funcione, caso não tenha pilhas em casa, é ligar o circuito elétrico direto na placa Arduino, Figura 73. O Arduino possui um regulador de tensão de 5 volts que pode ser alimentado pela porta USB ou conectado a uma fonte de energia, que pode variar entre 7 a 12 volts, Figura 74. Essa energia alimenta os pinos de saída da placa, como, por exemplo, os pinos 5V, 3.3V e os pinos de 0 a 13.

Outro pino importante é o GND (ground ou terra), que serve como o negativo (catodo) do circuito elétrico. Ligue uma ponta de um dos jumpers no pino GND e outra ponta do outro jumper no pino de saída 5V da placa Arduino. Caso a luzinha não acenda, lembre-se de que é só trocar a ordem da polaridade dos fios, trocar a ponta dos pinos da placa Arduino.

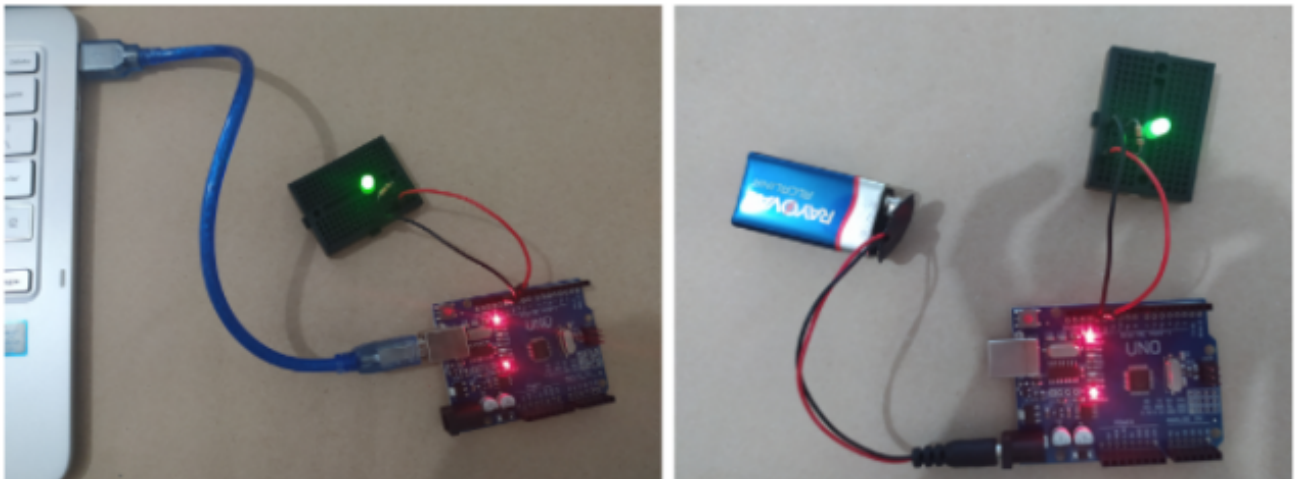
<sup>56</sup> Disponível em: <<https://medium.com/como-programar-em-1-dia/como-aprender-rob%C3%B3tica-em-1-dia-1a3edd04db7e>>. Acesso em 05/2022.

**Figura 73 - Circuito elétrico ligado no pino GND e no 5V do Arduino**



Fonte: Página medium.com<sup>57</sup>.

**Figura 74 - Circuito elétrico alimentado direto via USB e outro na fonte**



Fonte: Página medium.com.

Programa ou código LED com tempo programado

O objetivo desse circuito elétrico é a apresentação do funcionamento de um LED que liga e desliga em um intervalo de tempo.

Copie e cole na janela principal do editor de código, o programa “LED BLINK LED” que está descrito abaixo, também disponível em:

<https://www.tinkercad.com/things/9pgUoSeWPGD>

<sup>57</sup> Página medium.com, disponível em: <<https://medium.com/como-programar-em-1-dia/como-aprender-rob%C3%B3tica-em-1-dia-1a3edd04db7e>>. Acesso em 05/2022.

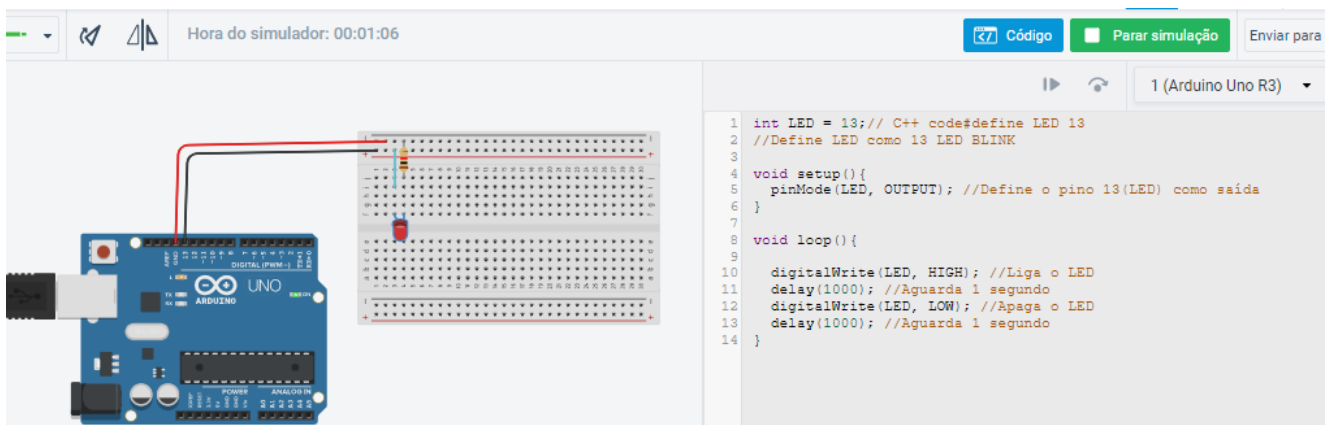
```

int LED = 13; // C++ code #define LED 13
//Define LED como 13 LED BLINK
void setup(){
  pinMode(LED, OUTPUT); //Define o pino 13(LED) como saída
}
void loop(){
  digitalWrite(LED, HIGH); //Liga o LED
  delay(1000); //Aguarda 1 segundo
  digitalWrite(LED, LOW); //Apaga o LED
  delay(1000); //Aguarda 1 segundo
}

```

Esse código do programa BLINK LED, Figura 75, fará a nossa placa Arduino piscar a luz LED que será conectada a ela. Na plataforma Tinkercad é possível simular e testar o código.

Figura 75 - Simulação do circuito Blink LED no Tinkercad



Fonte: Autoria própria 2022.

Acesse a esse código do BLINK LED em: <https://create.arduino.cc/editor/peters10/69e28a7f-a882-42e6-b782-67e309b52c0b/preview>.

As funções `pinMode()`, `digitalWrite()` e `delay()`, já são preenchidas automaticamente pelo compilador do Arduino. Em outra oficina didática, ainda teremos outras funções ou ainda diretivas como, por exemplo, as `#const` e `#define`, o que o modificador `#const` faz é “dizer” ao compilador que a variável (ou ponteiro) não pode ser alterada no decorrer do código.

A utilização do `#define`, que é uma diretiva do pré-processador do compilador que de certo modo torna mais simples a definição de uma variável estática que será usada múltiplas vezes, ou seja, no caso de ser necessário alterar o valor dessa variável estática basta mudar o valor do `#define` ao invés de andar a alterar todas as variáveis ao longo do código, sendo assim, é uma funcionalidade muito útil e poderosa.

A linha de código número 5: `pinMode(LED, OUTPUT);` a configuração que a porta de saída do programa será um número 13 pela declaração da variável na linha 1 `int LED = 13;`.

O conteúdo do método `loop()`, das linhas 8 a 14, instrua a placa a:

1. Envie uma quantidade alta de energia para a porta 13:  
`digitalWrite(LED, HIGH);`
2. Depois aguardar uma pausa de 1000 milissegundos (ou seja, 1 segundo):  
`delay(1000);`
3. Depois enviar baixa quantidade de energia para a porta 13:  
`digitalWrite(LED, LOW);`
4. Aguardar novamente mais 1000 milissegundos:  
`delay(1000);`

E ficar repetindo esses 4 passos indefinidamente.

Verificar e salvar.

Clique agora no botão “Verify and Save” para confirmar que está tudo certo com o seu novo programa.

Carregar programa para a placa Arduino.

Se tudo estiver certo com o seu programa, e ele tiver sido selecionado com sucesso, você poderá enviá-lo para a sua placa para que ela funcione.

Com sua placa Arduino devidamente conectada em seu computador pelo cabo USB, clique no botão “Upload and Save” do editor.

Aguarde o upload ser realizado e observe as reações da luzinha intermitente em sua placa.

Se todas as conexões estiverem certas (com o jumper positivo ligado na placa de saída número 13) a luzinha de LED se acenderá e começará a piscar em intervalos de 1 segundo (1000 milissegundos).

Parabéns! Seu circuito elétrico está funcionando agora com inteligência embutida.

Customizando o código

Agora volte ao seu código no editor online, altere o intervalo de tempo de intermitência da luzinha para 3 segundos (3000).

E também troque o pino de saída do 13 para o número 10.

Carregue esse novo código para sua placa Arduino e teste o resultado novamente!

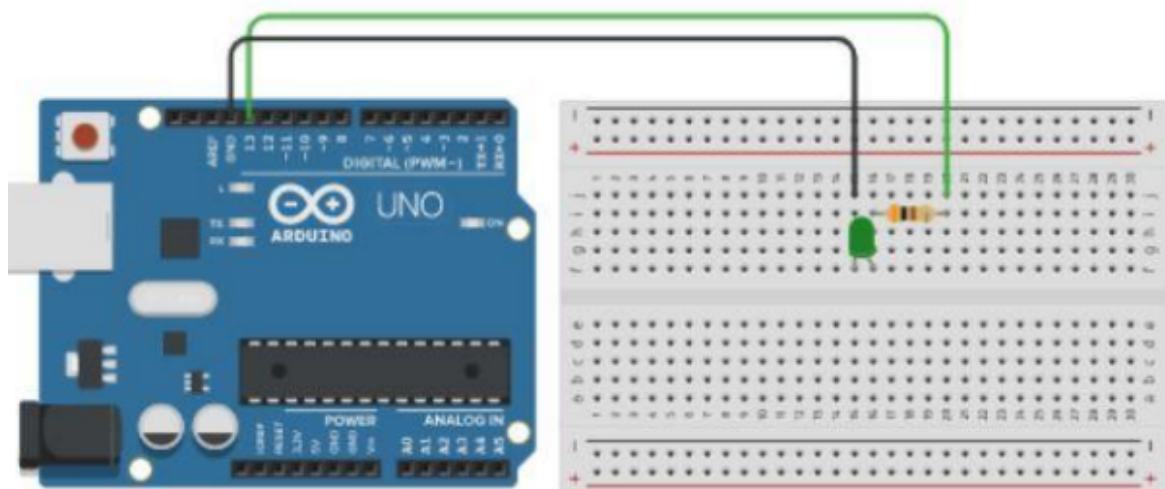
#### 6.1.4.3 Oficina didática - LED com tempo programado na prática.

Agora faça o circuito da oficina didática - Programando LED Blink na prática com os materiais físicos a seguir:

Materiais:

1. um kit da plataforma Arduino;
2. um notebook/desktop para cada aluno ou grupo de alunos;
3. uma protoboard;
4. um cabo USB;
5. fios jumper;
6. um resistor;
7. um LED.

Figura 76 - LED com tempo programado na prática



Fonte: Autoria própria.

Em posse dos materiais, você deve conectar os componentes conforme Figura 76, em seguida carregar o programa para a placa arduino. Por fim, verificar se está funcionando corretamente.

Caso queira, pode-se conectar a Placa Arduino numa fonte 5v e deixar o programa rodando normalmente.

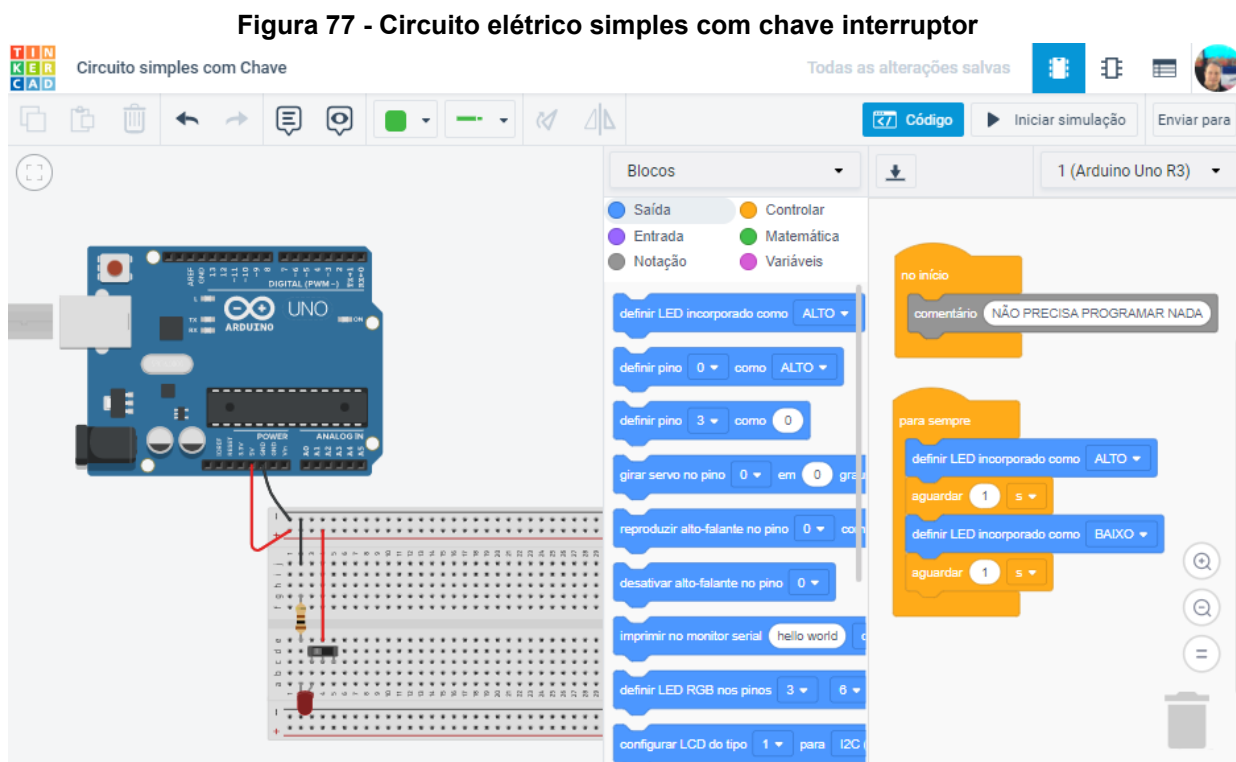
#### 6.1.4.4 Oficina didática - Um LED com interruptor

O objetivo desta oficina didática é apresentar o funcionamento de um LED que pode ser controlado por uma chave ou interruptor, Figura 77.

Materiais:

1. uma placa Arduino UNO;
2. um notebook/desktop para cada aluno ou grupo de alunos;
3. uma protoboard;
4. um cabo USB;
5. fios jumper;
6. um resistor;
7. um LED;
8. uma chave interruptor deslizante.

Como construir um circuito elétrico simples com uma chave interruptor?



Fonte: Autoria própria.

Para montar esse projeto no **Tinkercad** e fazer funcionar não precisa programar. Basta ligar os componentes conforme a imagem e depois iniciar a simulação que irá funcionar normalmente.

Para que esse sistema funcione no meio físico faça as ligações conforme o esquema da imagem e acione a chave interruptor para poder verificar. Será que vai funcionar corretamente?

Agora é com você!

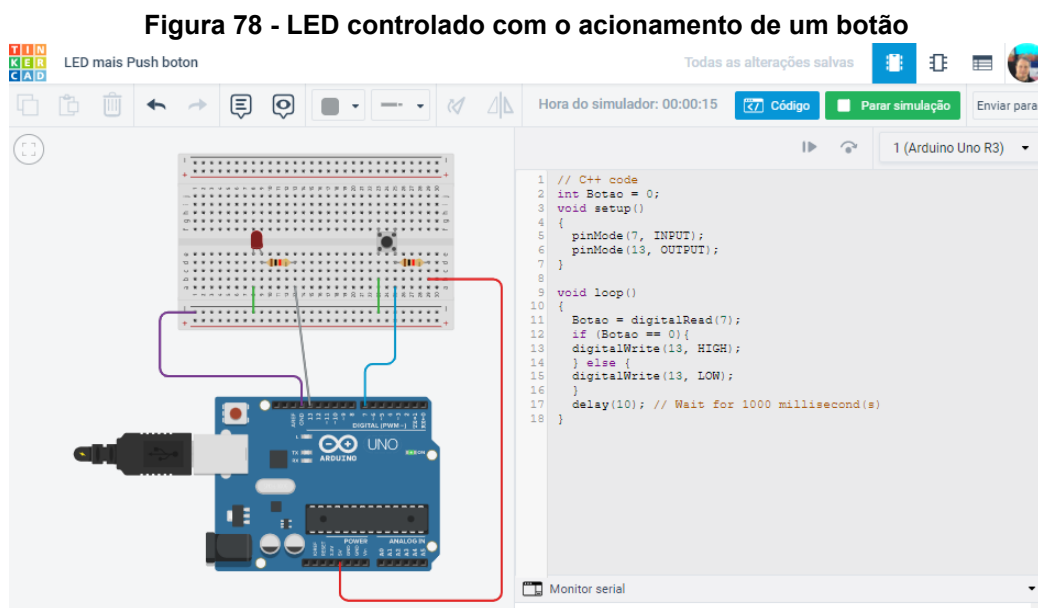
#### 6.1.4.5 Oficina didática - LED controlado com o acionamento de um botão

Esta é uma oficina didática bastante similar a anterior cujo objetivo é apresentar o funcionamento de um LED controlado com o acionamento de um botão para acender ou apagar o LED.

Materiais:

1. um Arduino;
2. uma protoboard;
3. um cabo USB;
4. fios jumper;
5. dois resistores;
6. um LED;
7. um Push botão;
8. um notebook/desktop para cada aluno ou grupo de alunos.

Representação do sketch do projeto, Figura 78.



Fonte: Autoria própria, Tinkercad 2022.



Momentos didáticos: Como controlar a luz de um LED utilizando um botão?  
Conceituação e resolução de como controlar a luz de um LED utilizando o acionamento do botão.

Como ligar LEDs em sequência utilizando um ou mais botões?

Código do projeto:

```
// C++ code
int Botao = 0;
  void setup()
  {
    pinMode(7, INPUT);
    pinMode(13, OUTPUT);
  }
  void loop()
  {
    Botao = digitalRead(7);
    if (Botao == 0){
      digitalWrite(13, HIGH);
    } else {
      digitalWrite(13, LOW);
    }
    delay(10); // Wait for 10 millisecond(s)
  }
}
```

Então, quais dessas duas oficinas didáticas (oficina didática “Um LED com Interruptor” ou “LED controlado com acionamento de um botão”) você achou mais fácil? Por quê?

#### 6.1.4.6 Oficina didática - 2 LEDs - Sinalizador de garagem

O objetivo dessa oficina didática é apresentar o funcionamento de dois LEDs simultâneos, simulando um Sinalizador de Garagem<sup>58</sup>, Figura 79.

Materiais:

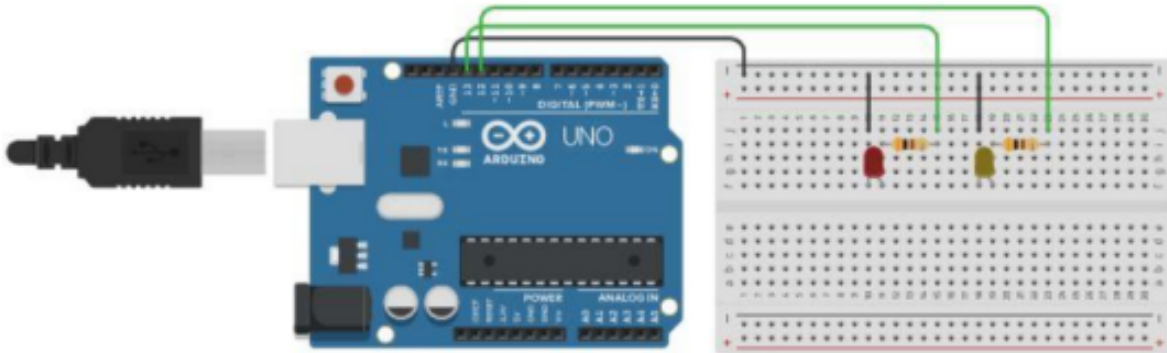
1. um kit da plataforma Arduino;
2. um notebook/desktop para cada aluno ou grupo de alunos;
3. uma protoboard;
4. um cabo USB;

---

<sup>58</sup> Adaptado de: <[https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES\\_.20.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES_.20.2020-1.pdf)> pág. 14. Acesso em 03/2022.

5. fios jumper;
6. dois resistores;
7. dois LEDs.

Figura 79 - Sketch do desafio sinalizador de garagem



Fonte: Autoria própria, Tinkercad 2022.

Momentos didáticos:

Como desenvolver um sinalizador de garagem?

Conceituação e resolução de um sinalizador de garagem.

Código do Projeto:

```
const int pinoLedV = 13; //define pino led vermelho
const int pinoLedA = 12; //define pino led amarelo
void setup(){
  pinMode(pinoLedV, OUTPUT); //define o pino como saída
  pinMode(pinoLedA, OUTPUT); //define o pino como saída
}
void loop() {
  digitalWrite(pinoLedV, HIGH); //liga o led vermelho
  digitalWrite(pinoLedA, LOW); //desliga o led amarelo
  delay(500); //define o tempo para que um led apague e outra acenda
  digitalWrite(pinoLedA, HIGH); //liga o led amarelo
  digitalWrite(pinoLedV, LOW); //desliga o led vermelho
  delay(500); //define o tempo para que um led apague e outra acenda
}
```

Essa mesma oficina didática com código em Blocos (Acender 2 LEDs).

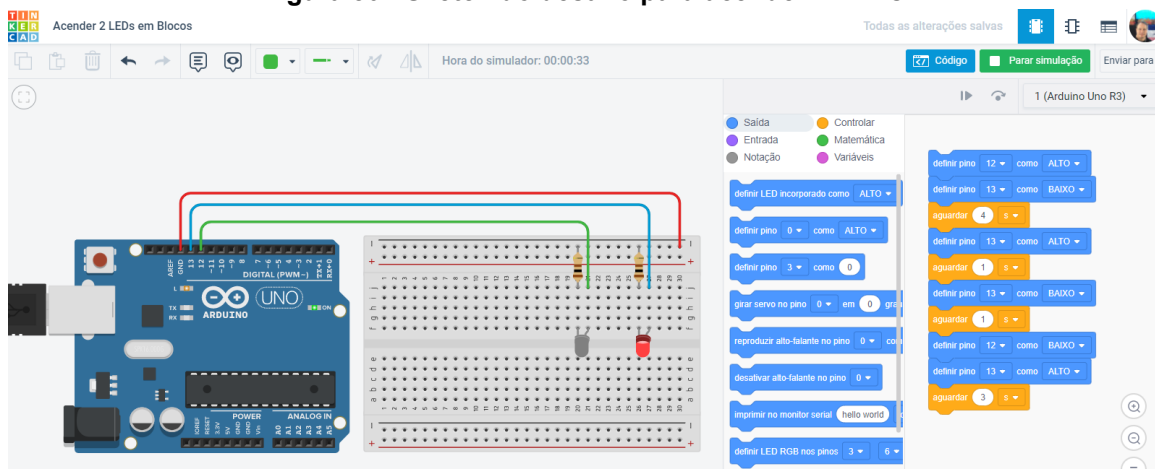
O objetivo dessa oficina didática é apresentar o funcionamento de dois LEDs de forma alternadas, Figura 80.

Materiais:

1. um kit com placa Arduino;
2. dois LEDs;
3. uma protoboard;
4. um cabo USB;
5. fios jumper;
6. dois resistores;
7. um notebook/desktop para cada aluno ou grupo de alunos.

Primeiro iremos montar na plataforma Tinkercad o esquema no exemplo da imagem abaixo:

**Figura 80 - Sketch do desafio para acender 2 LEDs**



Fonte: Autoria própria. Disponível em: Tinkercad<sup>59</sup>.

#### 6.1.4.7 Oficina didática - Cinco LEDs simulando um semáforo

O objetivo dessa oficina didática é apresentar o funcionamento de cinco LEDs simultâneos, simulando um semáforo que você mesmo pode construir.

Materiais:

1. um kit Arduino;
2. um computador para cada aluno ou grupo;
3. uma protoboard;
4. um cabo USB;
5. fios jumper;
6. cinco resistores;
7. cinco LEDs.

<sup>59</sup> Disponível em: <<https://www.tinkercad.com/things/l1X8OdYUZBP?sharecode=jqcC72pqBqUrKwJa-e-1O5Bz4CgXwXwUYbvHIZPMva98>>. Acesso em 03/2022.

Como construir um semáforo de carros e pedestres utilizando 5 LEDs, a partir do sketch abaixo? Fica como proposta de atividade para o leitor fazer a imersão.

Código ou sketch:

```
//define as variáveis dos pinos que estão conectados aos LEDs
const int carroVermelho = 13;
const int carroAmarelo = 12;
const int carroVerde = 11;
const int pedestreVermelho = 10;
const int pedestreVerde = 9;
    void setup(){ //define as variáveis dos leds como saída
pinMode(carroVermelho, OUTPUT);
pinMode(carroAmarelo, OUTPUT);
pinMode(carroVerde, OUTPUT);
pinMode(pedestreVermelho,OUTPUT);
pinMode(pedestreVerde, OUTPUT);
}
    void loop() {
//define a sequência, situação dos leds e tempo para que funcione
como um semáforo
    digitalWrite(carroVerde, HIGH);
    digitalWrite(pedestreVermelho, HIGH);
    delay(5000);
    digitalWrite(carroVerde, LOW);
    digitalWrite(carroAmarelo, HIGH);
    delay(3000);
    digitalWrite(carroAmarelo, LOW);
    digitalWrite(carroVermelho, HIGH);
    digitalWrite(pedestreVermelho, LOW);
    digitalWrite(pedestreVerde, HIGH);
    delay(5000);
    digitalWrite(carroVermelho, LOW);
    digitalWrite(pedestreVerde, LOW);
}
```

Fonte: Oficina adaptada de: <[https://www.ufsm.br/app/uploads/sites/517/2021/03/PP\\_GTER.DES\\_.20.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PP_GTER.DES_.20.2020-1.pdf)>. pg 14. Acesso em 03/2022.

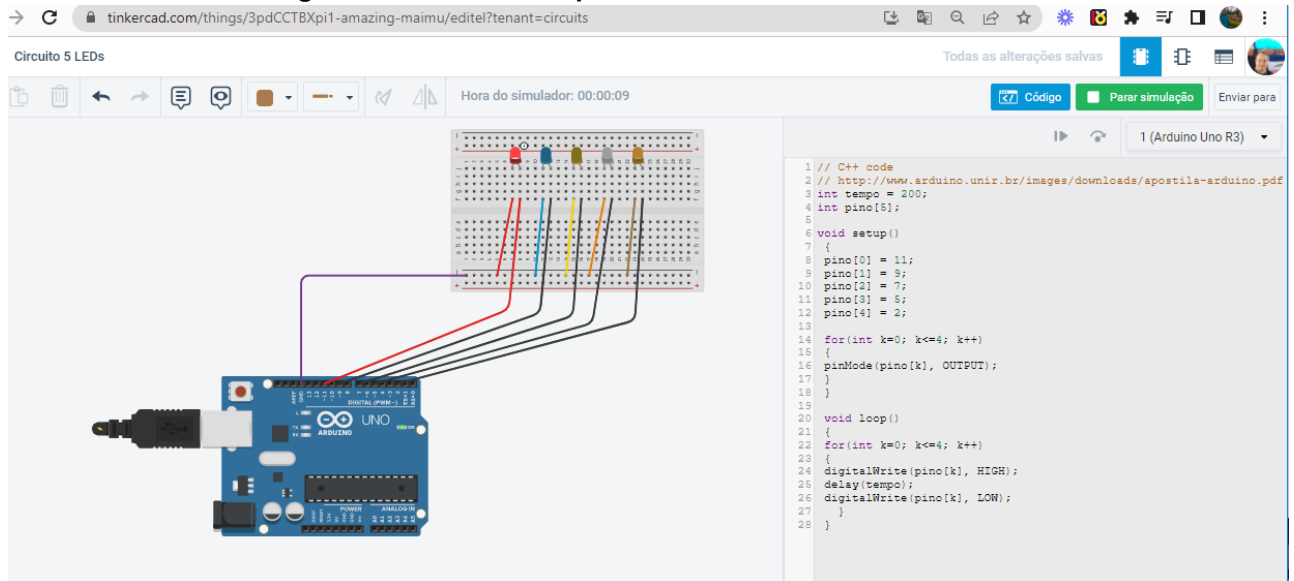
#### 6.1.4.8 Oficina didática - Cinco LEDs piscando de forma alternada

Escrevendo o código e definindo com 200 ms (milissegundos) o tempo no qual cada LED ficará aceso, podemos ter no código a seguinte linha de comando: `int tempo = 200;` os números das portas serão armazenados em um vetor de 05 componentes inteiras de forma que colocaremos no código o comando `int pino[5]`. Para fazermos os LEDs acenderem alternadamente, Figura 81, colocaremos os comandos `digitalWrite(pino[k], HIGH);` e `digitalWrite(pino[k], LOW);` dentro de um loop `for`, separados pela chamada da função `delay()`.

Veja o código completo:

```
int tempo = 200;
int pino[5];
void setup() {
  pino[0] = 11;
  pino[1] = 9;
  pino[2] = 7;
  pino[3] = 5;
  pino[4] = 2;
  for(int k=0; k<=4; k++) {
pinMode(pino[k], OUTPUT);
  }
}
void loop() {
  for(int k=0; k<=4; k++)
  {
digitalWrite(pino[k], HIGH);
delay(tempo);
digitalWrite(pino[k], LOW);
  }
}
```

**Figura 81 - Cinco LEDs piscando de forma alternada**



Fonte: Autoria própria feito no Tinkercad.

Agora faça você mesmo:

Nesse projeto fizemos os LEDs acenderem alternadamente, mas apenas em um sentido. Modifique o código para observar o pulso de luz percorrer o LED nos dois sentidos de forma alternada, isto é, primeiro da esquerda para a direita, depois da direita para a esquerda, e assim por diante.

Fonte: Oficina adaptada de: <<http://www.arduino.unir.br/images/downloads/apostila-arduino.pdf>>. (p.69). Acesso em 09/2022.

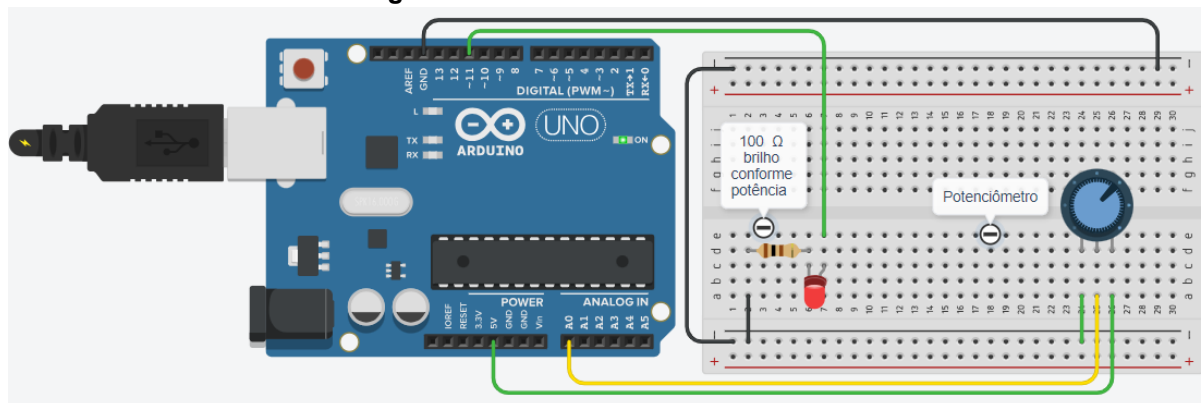
#### 6.1.4.9 Oficina didática - Potenciômetro controlando um LED

Nosso objetivo nessa oficina didática é apresentar o funcionamento de um LED cuja intensidade é controlada através de um potenciômetro. Representação do sketch do projeto, Figura 82.

Materiais:

1. um Arduino;
2. uma protoboard;
3. um cabo USB;
4. fios jumper;
5. um resistor;
6. um LED;
7. um potenciômetro;
8. um notebook/desktop para cada aluno ou grupo de alunos.

Figura 82 - Potenciômetro e o Arduino



Fonte: Plataforma UFSM<sup>60</sup>.

Link no Tinkercad:

<https://www.tinkercad.com/things/4iXgU02zE5D-sizzling-jaiks/editel?tenant=circuits>

Momentos didáticos:

Como controlar a potência de um LED utilizando potenciômetro?

Conceituação e resolução de como controlar a potência de vários LEDs utilizando potenciômetro, Figura 83.

Como ligar LEDs em sequência utilizando potenciômetro?

Observação: Nesta oficina iremos utilizar a função `map( )` que permite efetuar o mapeamento de um intervalo numérico, que usa números inteiros e não irá gerar números fracionários quando a matemática indicar que deveria. Números fracionários são truncados e não são arredondados. Usaremos também a função `analogWrite( )` que permite escrever um valor analógico (um sinal PWM) em um pino. É preciso chamar `pinMode( )` para configurar um pino como saída antes de chamar `analogWrite( )`.

Código ou sketch:

```
const int potPin = A0; //define o pino analógico do potenciômetro
const int ledPin = 11; //define o pino do led
float valPot = 0; //variável armazena o valor do potenciômetro
void setup(){
  pinMode(ledPin,OUTPUT); //define o pino como saída
  pinMode(potPin,INPUT); //define o pino como entrada
}
```

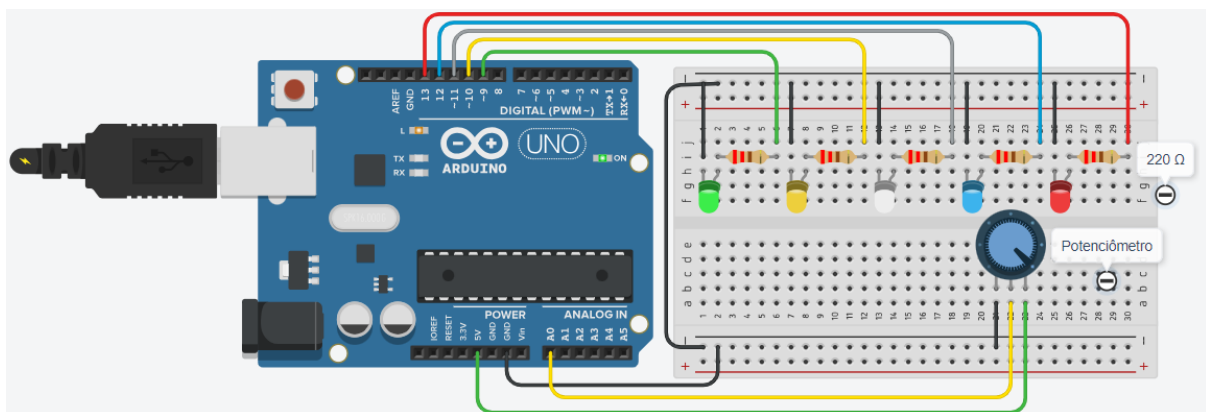
<sup>60</sup> Referência: Adaptado de Oficinas de arduino. Disponível em: <[https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES\\_.20.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES_.20.2020-1.pdf)>. Acesso em 03/2022.

```

void loop() {
  valPot = analogRead(potPin); //faz a leitura analógica do
potenciômetro
  valPot = map(valPot,0,1023,0,255); //faz o mapeamento do
potenciômetro para transformar em valores digitais PWM
  analogWrite(ledPin,valPot ); //aciona o led conforme valor do
potenciômetro
}

```

Figura 83 - Potenciômetro controlando LEDs



Fonte: Autoria própria, Tinkercad 2022.

Link no Tinkercad: <https://www.tinkercad.com/things/2VT8rZaInVK-funky-turing-hang-o/editel?tenant=circuits>

Código com a resposta do desafio:

```

const int potPin = A0; //define o pino analógico do potenciômetro
const int quantLeds = 5; //define a quantidade de leds
int ledPins[] = {9,10,11,12,13};
//define em um vetor os pinos que os leds estão ligados
void setup(){
  for(int i = 0;i<quantLeds;i++) //define os pinos dos leds como
saida
  {
    pinMode(ledPins[i],OUTPUT);
  }
}
void loop (){
  int valPoten = analogRead(potPin); //faz leitura do potenciômetro

```



```

int ledNivel = map(valPoten,0,1023,0,quantLeds);
//faz o mapeamento dos leds para o potenciômetro
for(int i = 0;i<quantLeds;i++)//percorre o vetor de leds
{
  if (i<ledNivel)
  {
digitalWrite(ledPins[i],HIGH);//acende o led
  }
  else {
digitalWrite(ledPins[i],LOW);//desliga o led
  }
}
}
}

```

Fonte: Oficina didática adaptada de:

[https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES\\_.20.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES_.20.2020-1.pdf)

#### 6.1.4.10 Oficina didática - Motor com potenciômetro

Objetivos: Apresentação e funcionamento de um motor cuja velocidade é controlada por meio de um potenciômetro.

Materiais:

- |                    |   |
|--------------------|---|
| 1. um Arduino;     | 8. um potenciômetro;  |
| 2. uma protoboard; | 9. uma fonte de energia externa;                            |
| 3. um cabo USB;    | 10. um notebook/desktop para cada aluno ou grupo de alunos. |
| 4. fios jumper;    |   |
| 5. um motor;       |   |
| 6. um transistor;  |   |
| 7. um diodo;       |   |

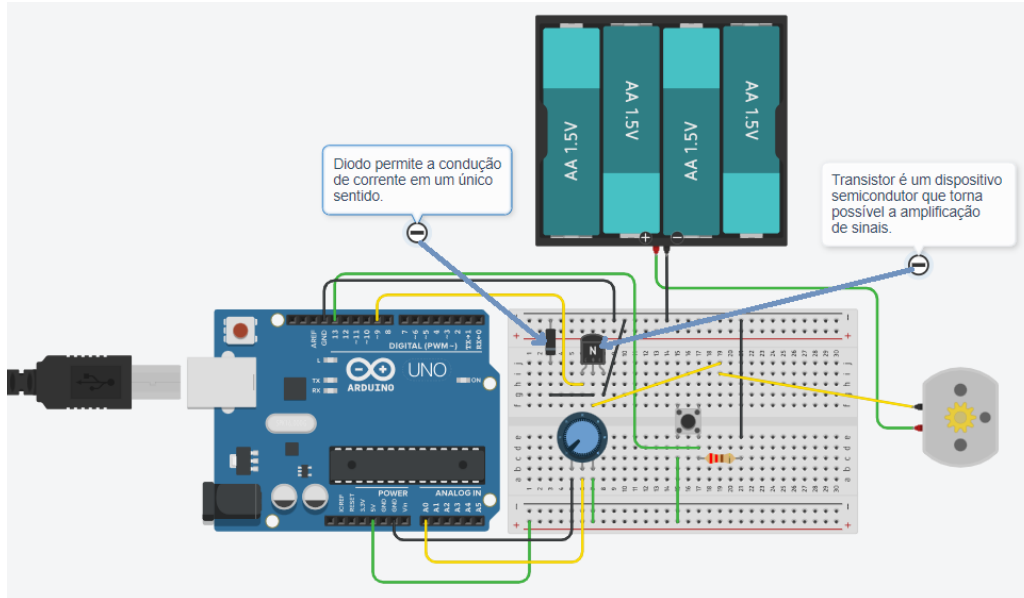
Momentos didáticos:

Como controlar a velocidade de um motor utilizando potenciômetro?

Conceituação e resolução de como controlar a velocidade de um motor utilizando potenciômetro, Figura 84.

Como controlar a velocidade, ligar e desligar um motor por meio de um botão?

Figura 84 - Motor com potenciômetro



Fonte: Autoria própria, Tinkercad 2022<sup>61</sup>.

Código com a resposta do projeto<sup>62</sup> motor com potenciômetro:

```
const int pot = A0; //define pino que está conectado o
potenciômetro
const int transistor = 9; //define pino que está conectado o
transistor
int valPot = 0; //armazena o valor do potenciômetro
const int botao = 13;
void setup() {
  pinMode(botao,INPUT); //define o pino como entrada
  pinMode(pot, INPUT); //define o pino como entrada
  pinMode(transistor, OUTPUT); //define o pino como saída
}
void loop() {
  if(digitalRead(botao) == HIGH) { //ativa o valor do
potenciômetro
    valPot = analogRead(pot); //variável recebe valor do
potenciômetro
    valPot = map(valPot,0,1023,0,255); //converte o valor do
potenciômetro em valores pwm
    analogWrite(transistor, valPot); //envia o valor do
potenciômetro ao motor
  } }
```

<sup>61</sup>Link no Tinkercad: <https://www.tinkercad.com/things/czuzAoDqSYI-motor-com-potenciometro/editel>.

<sup>62</sup>Fonte: Oficina adaptada de: <[https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES\\_2.0.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES_2.0.2020-1.pdf)>. Acesso em 02/22.

#### 6.1.4.11 Oficina didática - Sensor de Luminosidade para acender uma lâmpada

O objetivo dessa oficina didática é apresentar o funcionamento de um fotoresistor, conhecido pela sigla inglesa LDR, é um componente eletrônico passivo do tipo resistor variável dependendo da intensidade da luz que incide sobre o LDR que nesse caso, ao não identificar a presença de luz, o LED acende.

Materiais:

1. um Arduino;
2. uma protoboard;
3. um cabo USB;
4. fios jumper;
5. dois resistores (1 de 100 ohm outro de 220 ohm);
6. um LED (usar resistor 220 ohm);
7. um fotoresistor (utilizar o resistor 100 ohm);
8. um Buzzer (para completar o desafio);
9. um notebook/desktop para cada aluno ou grupo de alunos.

Momentos didáticos:

Como detectar luz em um ambiente e informar através de um LED?

Conceituação e resolução de como detectar luz em um ambiente, Figura 85.

Desafio:

Detectar luz em um ambiente e informar através de som se tiver luz ou não.

Código do projeto, adaptado de: “Oficinas<sup>63</sup> de Arduino: Manual para professores de lógica de programação.”

```
int ledPin = 13; //define pino digital do led
int ldrPin = A0; //define pino analogico do LDR
int ldrValor = 0; //variável que armazena valor do sensor
void setup() {
    pinMode(ledPin,OUTPUT); //define o pino do led como saída
    Serial.begin(9600); //inicializa a comunicação serial
}
void loop() {
    ldrValor = analogRead(ldrPin); //armazena valor do
sensor na variável
    if (ldrValor >= 800){
```

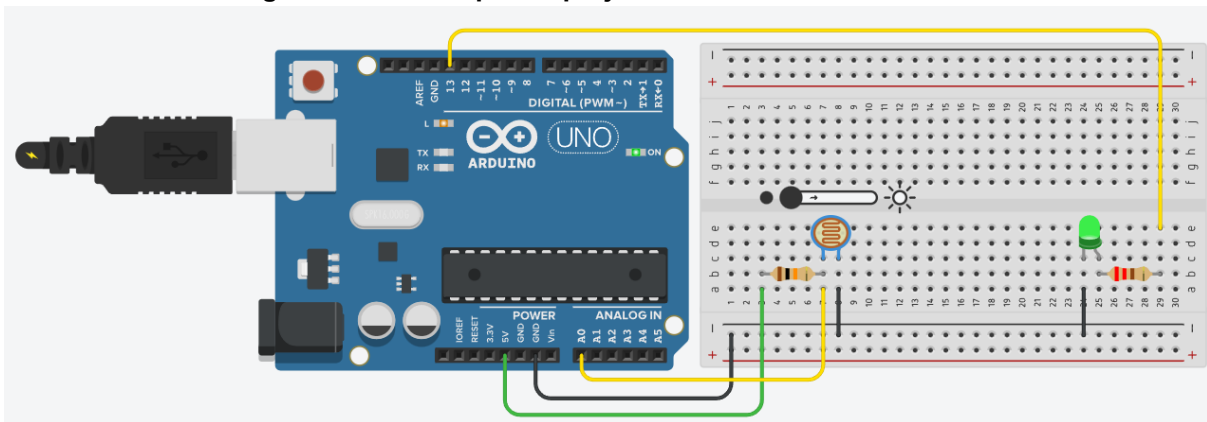
<sup>63</sup>Adaptado de: Oficinas de Arduino: Manual para professores de lógica de programação. Disponível em: <[https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES\\_.20.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES_.20.2020-1.pdf)>. Acesso em 03/2022.

```

        //se o valor do sensor for maior que 800 liga o
led
    digitalWrite(ledPin,HIGH);//define o led como ligado
    }
    else{
    digitalWrite(ledPin,LOW);//define o led como desligado
    }
    Serial.println(ldrValor);//informa o valor do sensor na tela
    delay(100);
}

```

Figura 85 - Sketch para o projeto sensor de luminosidade



Fonte: Autoria própria, Tinkercad 2022<sup>64</sup>.

Código resposta do Desafio, sketch na Figura 86:

```

int ledPin = 13; //define pino digital do led
int buzzerPin = 12;//define pino digital do buzzer
int ldrPin = A0; //define pino analogico do led
int ldrValor = 0; //variável que armazena valor do sensor
void setup() {
    pinMode(ledPin,OUTPUT); //define o pino do led como saída
    pinMode(buzzerPin,OUTPUT);//define o pino do buzzer como saída
    Serial.begin(9600); //inicializa a comunicação serial
}
void loop() {
    ldrValor = analogRead(ldrPin); //armazena valor do sensor na
variável
    if (ldrValor >= 800){
        //se o valor do sensor for maior que 800 liga o led

```

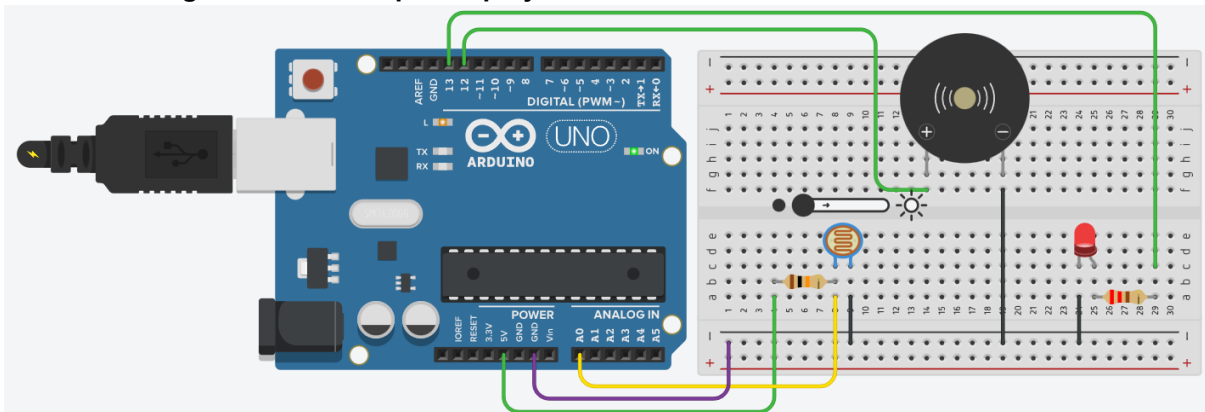
<sup>64</sup>Link no Tinkercad: <https://www.tinkercad.com/things/6lqGNnmv4CP-sensor-de-luminosidade-ldr/edi> tel.

```

digitalWrite(ledPin,HIGH);//define o led como ligado
digitalWrite(buzzerPin,HIGH);//define o buzzer como ligado
    }
else {
digitalWrite(ledPin,LOW);//define o led como desligado
digitalWrite(buzzerPin,LOW);//define o buzzer como desligado
    }
Serial.println(ldrValor);//informa o valor do sensor na tela
}

```

Figura 86 - Sketch para o projeto sensor de luminosidade mais buzzer



Fonte: Autoria própria, Tinkercad 2022<sup>65</sup>.

Fonte: Oficina adaptada de: <[https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES\\_.20.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES_.20.2020-1.pdf)>. Acesso em 01/22.

#### 6.1.4.12 Oficina didática - Alarme de segurança com sensor infravermelho

Esta oficina didática utiliza o Arduino detectando presença com sensor infravermelho tem por objetivo apresentar o funcionamento de um sensor de movimento, Figura 87, que ao identificar um movimento no ambiente um LED acende, como uma luz em um corredor, por exemplo.

O sensor infravermelho é um dispositivo eletrônico que emite e/ou detecta radiação infravermelha com o intuito de revelar alguns aspectos do ambiente ao seu redor. Esses sensores podem ser utilizados para detectar movimentos, alarmes de passagem, medir o calor de um objeto, em leitores de código de barras, entre outros.

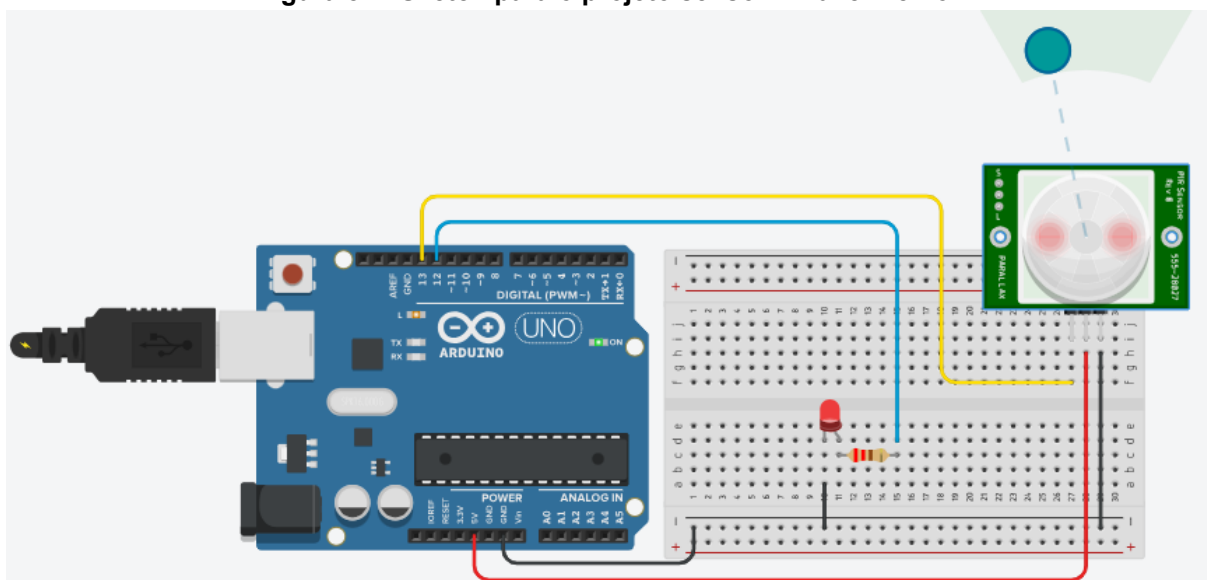
<sup>65</sup> Link no Tinkercad: <https://www.tinkercad.com/things/jWMQ9hSxC63-sensor-de-luminosidade/editel>.

Os sensores infravermelhos, existem vários tipos e modelos, são muito utilizados, seja como detector de movimento, na robótica educacional para robôs seguidor de linha, como leitores de códigos de barras, ou outros.

Materiais:

1. um Arduino;
2. uma protoboard;
3. um cabo USB;
4. fios jumper;
5. um resistor;
6. um LED;
7. um sensor movimento;
8. um Buzzer. (para o desafio);
9. um notebook/desktop para cada aluno ou grupo de alunos.

Figura 87 - Sketch para o projeto sensor infravermelho



Fonte: Autoria própria, Tinkercad 2023<sup>66</sup>.

Momentos didáticos:

Como detectar presença com sensor infravermelho e informar situação por meio de LED?

Conceituação e resolução de como detectar presença com sensor infravermelho, sketch na Figura 88.

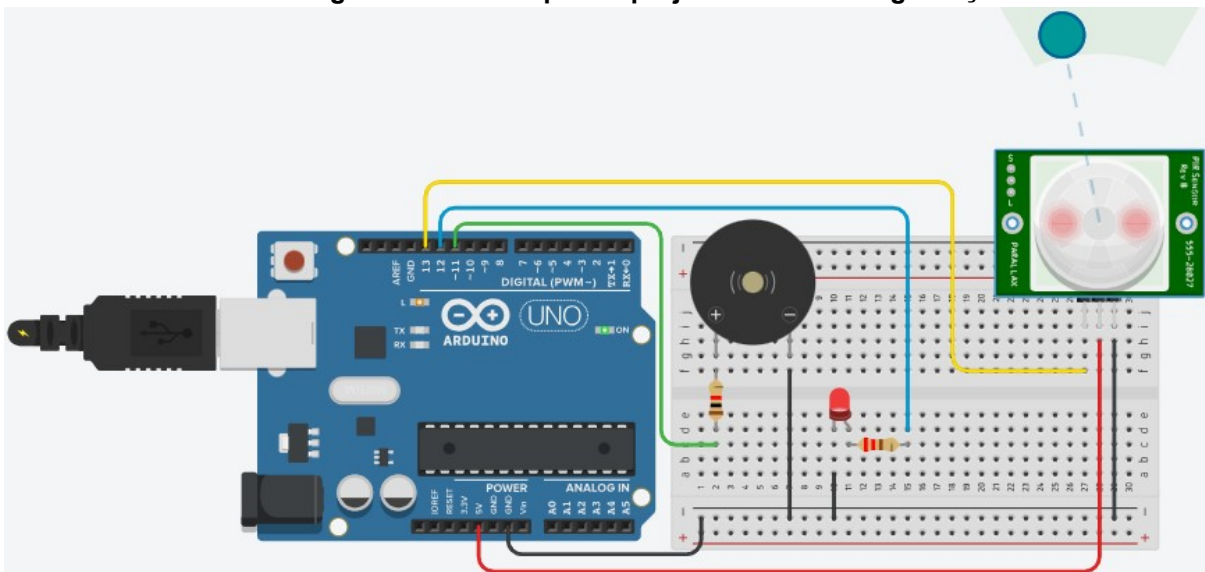
Como detectar presença com sensor infravermelho e informar presença por meio de som?

<sup>66</sup> Fonte: Acesse o sketch no link: <https://www.tinkercad.com/things/0fQdfVWGKkV-sensor-de-presenca/editel>.

Código do projeto.

```
const int pinoPIR = 13; //pino digital utilizado pelo sensor de
presença
const int pinoLED = 12; //pino digital utilizado pelo led
void setup(){
  pinMode(pinoLED, OUTPUT); //define o pino como saída
  pinMode(pinoPIR, INPUT); //define o pino como entrada
}
void loop(){
  if(digitalRead(pinoPIR) == HIGH){
    //se a leitura do pino for igual a high, faz
    {
      digitalWrite(pinoLED, HIGH); //acende o led
    }
  }
  else {
    digitalWrite(pinoLED, LOW); //apaga o led
  }
}
```

Figura 88 - Sketch para o projeto alarme de segurança



Fonte: Autoria própria, Tinkercad 2023.

Fonte: Adaptado de: <<https://www.tinkercad.com/things/2A585xdDTTb-sensor-de-presenca-com-buzzer/editel>>. Acesso em 06/2023.

Para quem desejar criar projetos mais robustos, como por exemplo para instalação de sistema de segurança, é indicado estudar as especificidades técnicas dos componentes para melhor eficiência na aplicação.

Resposta do Desafio do Projeto.

```
const int pinoPIR = 13; //pino digital utilizado pelo sensor de
presença
const int pinoLED = 12; //pino digital utilizado pelo led
const int pinoBUZZER = 11; //pino digital utilizado pelo Buzzer
void setup(){
  pinMode(pinoLED, OUTPUT); //define o pino como saída
  pinMode(pinoPIR, INPUT); //define o pino como entrada
  pinMode(pinoBUZZER, OUTPUT); //define o pino de saída
}
void loop()
{
  if(digitalRead(pinoPIR) == HIGH){
    //se a leitura do pino for igual a high, faz
    {
      digitalWrite(pinoLED, HIGH); //acende o led
      digitalWrite(pinoBUZZER, HIGH); //liga o buzzer
    }
  }
  else {
    digitalWrite(pinoLED, LOW); //apaga o led
    digitalWrite(pinoBUZZER, LOW); //desliga o buzzer
  }
}
```

Fonte: Adaptado de: <[https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES\\_.20.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES_.20.2020-1.pdf)>. Acesso em 06/2023.

#### 6.1.4.13 Oficina didática - Robô carro 2 WD

Esta oficina didática pretende desenvolver projetos mais amplos com base nas práticas exercitadas até aqui, aplicando conhecimentos adquiridos.

A proposta aqui é o desenvolvimento de um protótipo de um carro com funções autônomas. Existem várias possibilidades e aqui serão apresentadas algumas delas de forma bastante simplificada. Assim, fica a critério do professor escolher a que melhor convém, levando em conta materiais disponíveis, níveis de complexidade e necessidades do projeto desejado.



Materiais:

1. Arduino;
2. protoboard;
3. cabo USB;
4. fios jumper;
5. sensores de luminosidade;
6. resistores 470 ohms;
7. dois motores DC;
8. um notebook/desktop para cada aluno ou grupo de alunos.

Momentos Didáticos:

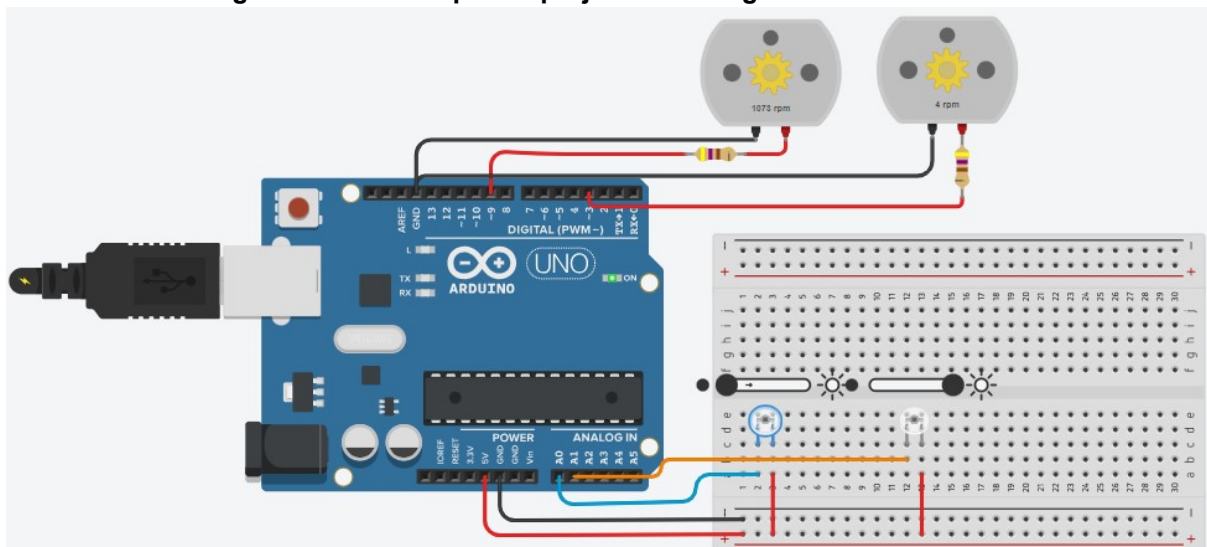
Como desenvolver o protótipo de um carro com funções autônomas?

Conceituação de como construir um protótipo de carro com funções autônomas.

Resolução de como construir e colocar em funcionamento o protótipo de um carro com funções autônomas.

Nesse primeiro momento veja como fica, na Figura 89, o carro seguidor de linha com dois motores DC e dois sensores de luminosidade LDR.

Figura 89 - Sketch para o projeto carro seguidor de linha 2WD



Fonte: Autoria própria, Tinkercad 2023.

Fonte: Adaptado de: <<https://www.tinkercad.com/things/3yTDVF2MqPI-copy-of-carri-cho-seguidor-de-linha/editel?tenant=circuits>>. Acesso em 06/2023.

Código do Projeto carro seguidor de linha com dois motores DC:

```

#define motor1 3
#define motor2 9
int val = 0;
int val1 = 0;
void setup(){
  pinMode(motor1, OUTPUT);
  pinMode(motor2, OUTPUT);
}
void loop(){
  val = analogRead(A0);
  val1 = analogRead(A1);
  if(analogRead(A0) < analogRead(A1)){
    analogWrite(motor1, val/4);
    analogWrite(motor2, val1/2);
  }
  else{
    analogWrite(motor1, val/2);
    analogWrite(motor2, val1/4);
  }
}
}

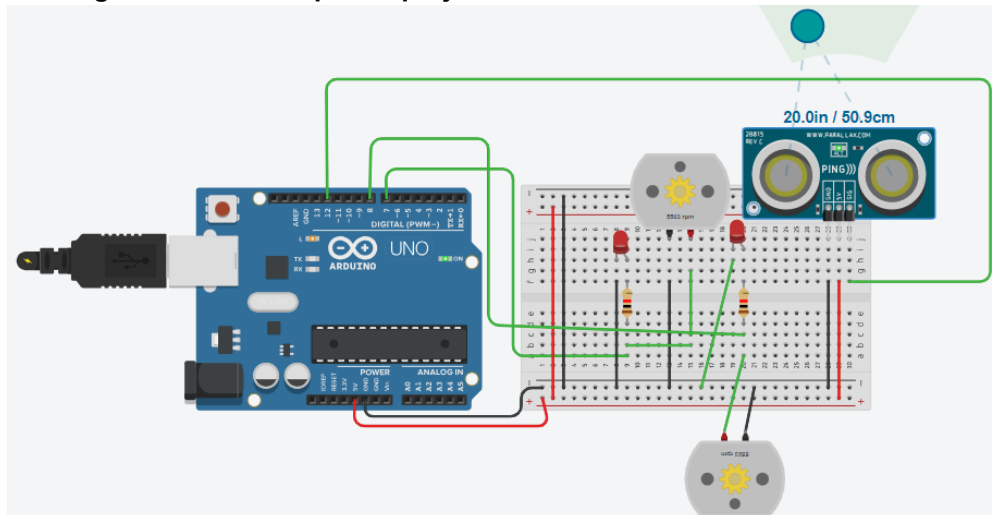
```

Neste próximo projeto, uma continuidade e aperfeiçoamento do projeto anterior, veja como fica o carro com dois motores DC que contorna obstáculos, utilizando o sensor de distância. Veja a Figura 90.

Materiais:

1. Arduino;
2. protoboard;
3. cabo USB;
4. fios jumper;
5. sensores de luminosidade;
6. resistores 220 ohms;
7. dois motores DC;
8. sensor de distância;
9. dois leds;
10. um notebook/desktop para cada aluno ou grupo de alunos.

Figura 90 - Sketch para o projeto carro contorno de obstáculos 2WD



Fonte: Autoria própria, Tinkercad 2023.

Código do Projeto contorno de obstáculos:

```
int tilt = 2; // define o número do pino de entrada
int led = 13; // define o número do pino de saída
int sigpin= 12;
int led2 = 7;
int led3 = 8;

void setup(){
  Serial.begin(9600);
  pinMode(tilt, INPUT);
  pinMode(led, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}

void loop(){
  int reading;
  reading = digitalRead(tilt);
  if(reading)
    digitalWrite(led, LOW);
  else
    digitalWrite(led, HIGH);
  long duration, cm;
  pinMode(sigpin, OUTPUT);
  digitalWrite(sigpin, LOW);
  delayMicroseconds(2);
  digitalWrite(sigpin, HIGH);
  delayMicroseconds(5);
```

```

digitalWrite(sigpin, LOW);
pinMode(sigpin, INPUT);
duration = pulseIn(sigpin, HIGH);
// converte o tempo em uma distância
cm = microsecondsToCentimeters(duration);
if(cm >=50){
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
    delay(200);
}
else if(cm <=5){
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    delay(200);
}
delay(100);
}
long microsecondsToCentimeters(long microseconds) {
    return microseconds / 29 / 2;
}

```

Fonte: Adaptado de: <<https://www.tinkercad.com/things/fZRNXHrOtwm-copy-of-robo-t-mobile/editel?tenant=circuits>>. Acesso em 06/2023.

Neste próximo projeto, veja como fica o carro com dois motores DC que segue linha, contorna obstáculos utilizando o sensor de distância que informa no LCD 16x2 a que distância há objetos à frente do sensor, Figura 91. O potenciômetro foi incluído no projeto com o objetivo de liberar a energia para o painel LCD 16X2 (2 linhas e 16 colunas) de forma a mostrar ou apagar o letreiro na tela. Para este projeto, ainda foi incluído no código da programação a biblioteca “LiquidCrystal.h” que permite que uma placa Arduino controle monitores LCD.

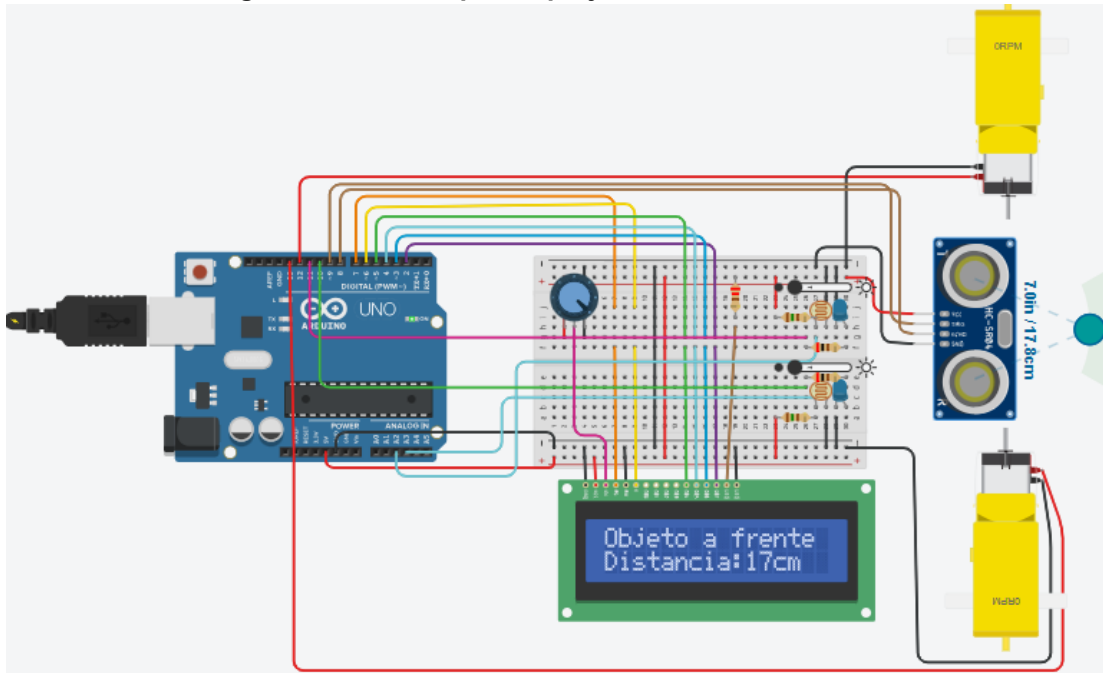
Materiais:

- |  |  |
|--|--|
| <ol style="list-style-type: none"> <li>1. Arduino;</li> <li>2. protoboard;</li> <li>3. cabo USB;</li> <li>4. fios jumper;</li> </ol> | <ol style="list-style-type: none"> <li>5. sensores de luminosidade;</li> <li>6. resistores 220 e 150 ohms;</li> <li>7. dois motores DC;</li> </ol> |
|--|--|

- 8. sensor de distância;
- 9. dois leds;
- 10. um potenciometro;
- 11. um painel LCD 16x2;

- 12. um notebook/desktop para cada aluno ou grupo de alunos.

Figura 91 - Sketch para o projeto carro 2WD com LCD



Fonte: Autoria própria, Tinkercad 2023.

Código do Projeto carro 2WD com LCD:

```
#include <LiquidCrystal.h>
int ldr1 = A2;      //Declaração de Variáveis
int ldr2 = A3;
int motor1 = 12;
int motor2 = 13;
int led1 = 11;
int led2 = 10;
int ldr1_ler;
int ldr2_ler;
const int trig = 9;
const int echo = 8;
long distancia;
long duracao;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
//Definindo pinos como saídas e entrada (echo)
void setup(){
```

```

Serial.begin(9600);
lcd.begin(16,2);
pinMode(trig, OUTPUT);
pinMode(echo, INPUT);
pinMode(motor1, OUTPUT);
pinMode(motor2, OUTPUT);
pinMode(led1, OUTPUT);
pinMode(led2, OUTPUT);
}
void loop(){
  ldr1_ler = analogRead(ldr1);
  ldr2_ler = analogRead(ldr2);
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);
  duracao = pulseIn(echo, HIGH); //pulseIn captura o valor do
pulso em um pino
  distancia = duracao * 0.017175;
  if (distancia < 335){ //Se houver um objeto dentro do raio
do sensor
  lcd.setCursor(0,0); //ele irá informar a distância no
display LCD
  lcd.print("Objeto a frente");
  lcd.setCursor(0,1);
  lcd.print("Distancia:");
  lcd.print(distancia);
  lcd.print("cm");
  delay(500);
  lcd.clear();
}
  if (distancia > 334){ //Se não houver um objeto no raio do
sensor ele irá
  lcd.setCursor(0,0); //informar no display LCD que não possui
objeto a frente
  delay(500);
  lcd.clear();
  lcd.print("Nenhum objeto");
}
  if (ldr1_ler > 900 && ldr2_ler > 900){ //Motor 1, Motor 2 e
ambos leds irão ligar

```

```

    digitalWrite(motor1,HIGH);
    digitalWrite(motor2,HIGH);
    digitalWrite(led1, 1);
    digitalWrite(led2, 1);
}
if (ldr1_ler > 900 && ldr2_ler < 900){ //Motor 1 irá ligar,
Motor 2 irá desligar
    digitalWrite(motor1,HIGH);           //e apenas o led 1
acenderá
    digitalWrite(motor2,LOW);
    digitalWrite(led1, 1);
    digitalWrite(led2, 0);
}
if (ldr1_ler < 900 && ldr2_ler > 900){ //Motor 2 irá ligar,
Motor 1 irá desligar
    digitalWrite(motor1,LOW);           //e apenas o led 2
acenderá
    digitalWrite(motor2,HIGH);
    digitalWrite(led1, 0);
    digitalWrite(led2, 1);
}
if (ldr1_ler < 900 && ldr2_ler < 900){ // Ambos os motores
serão desligados
    digitalWrite(motor1,LOW);           // e nenhum led
acenderá
    digitalWrite(motor2,LOW);
    digitalWrite(led1, 0);
    digitalWrite(led2, 0);
}
if (distancia < 30){ // Se detectado um objeto com uma
distância menor de 30cm
    digitalWrite(motor1, LOW); // o robô irá parar
    digitalWrite(motor2, LOW);
    digitalWrite(led1, 0);
    digitalWrite(led2, 0);
}
}
}

```

Fonte: Adaptado de: <<https://www.tinkercad.com/things/c2tb5NyBJx1-copy-of-robot-seguidor-de-linha-com-sensor-de-parada/editel?tenant=circuits>>. Acesso em 06/2023.

Já estamos com bastante material criado, fica aqui como proposta, estes que podem e devem ser considerados, principalmente para aqueles leitores que não tenham expertises em projetos no Tinkercad implementado-os e estendendo com mais componentes tanto em projetos físicos ou mesmo no simulador. Nesse sentido, fica como proposta/desafio para o leitor, que possa usar a imaginação e desenvolver novos projetos com inclusão de mais componentes eletrônicos e novas formas de pensar, por exemplo a inclusão de um controle remoto IR, proposta para aguçar a curiosidade e continuar a pesquisar sobre o assunto sempre implementando novos componentes e novas possibilidades.

Fonte: Oficina didática adaptado de: <[https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES\\_.20.2020-1.pdf](https://www.ufsm.br/app/uploads/sites/517/2021/03/PPGTER.DES_.20.2020-1.pdf)>. Acesso em 03/2022.

#### 6.1.5 Circuito de aprendizagem Aplicativos para celular

O objetivo desse circuito de aprendizagem é apresentar ao professor algumas possibilidades em desenvolver habilidades para o uso e criação de APP de celular para incorporar em projetos que requerem um pouco mais de recursos.

##### 6.1.5.1 Oficina didática - App Inventor

O objetivo dessa oficina didática é possibilitar ao professor e ao estudante o desenvolvimento de habilidades e competências para trabalhar com o MIT App Inventor e assim, aplicar esse conhecimento adquirido em seus projetos diversos.

O MIT App Inventor, Figura 92, é um ambiente de programação visual intuitivo que permite criação de aplicativos totalmente funcionais para smartphones e tablets Android e IOS. É uma ferramenta baseada em blocos que facilita a criação de aplicativos complexos e de alto impacto em muito menos tempo, os novos aplicativos do MIT App Inventor podem ser instalados com tudo funcionando em menos de 30 minutos. E mais, o projeto MIT App Inventor visa democratizar o desenvolvimento de software ao capacitar todas as pessoas, especialmente os jovens, para passar do consumo de tecnologia à criação de tecnologia. Criada no MIT - Massachusetts Institute of Technology, é uma ferramenta bastante fácil de utilizar, até mesmo por quem não tem experiência em programação.

Vamos começar?



A primeira coisa a fazer é conhecer o ambiente, acesse o link <https://appinventor.mit.edu/>, já nessa página clique em “Crie aplicativos!”.



Fonte: Ambiente MIT APP Inventor<sup>67</sup>.

Feito isso, crie uma conta ou utilize um e-mail Google para iniciar (Observação: e-mail Google obrigatório). A partir daí, você será direcionado para o ambiente de desenvolvimento e já pode dar início aos trabalhos, Figura 93.

O nosso primeiro aplicativo teste padrão para todo programador, é o “Hello World!”. Explique para os estudantes as características de aplicativos implementados no App Inventor: - São criados com componentes que podem ser visíveis ou invisíveis - Os componentes têm propriedades (como cor, altura, largura) - Os componentes têm comportamentos que definem como ele deve responder a eventos ou iniciar suas atividades. Há dois modos de trabalhar Designer, para montar a interface gráfica, e o modo Blocks onde implementamos o funcionamento e a lógica do aplicativo. Peça para os alunos procurarem na tela os botões “Designer” ou “Projetista” e “Blocks” ou “Blocos” (geralmente fica no canto superior direito da tela) para ver o que acontece quando clicamos neles. Acesse o site (<http://appinventor.mit.edu/explore/>) em seguida clicar em “Create apps” ou “Crie aplicativos”, depois em “Start New Project” ou “Iniciar novo projeto”, e daí nomear o projeto (nome pode ser escolhido como queira ou ainda colocar “Hello World”). Assim que der ok, irá aparecer o ambiente de trabalho.

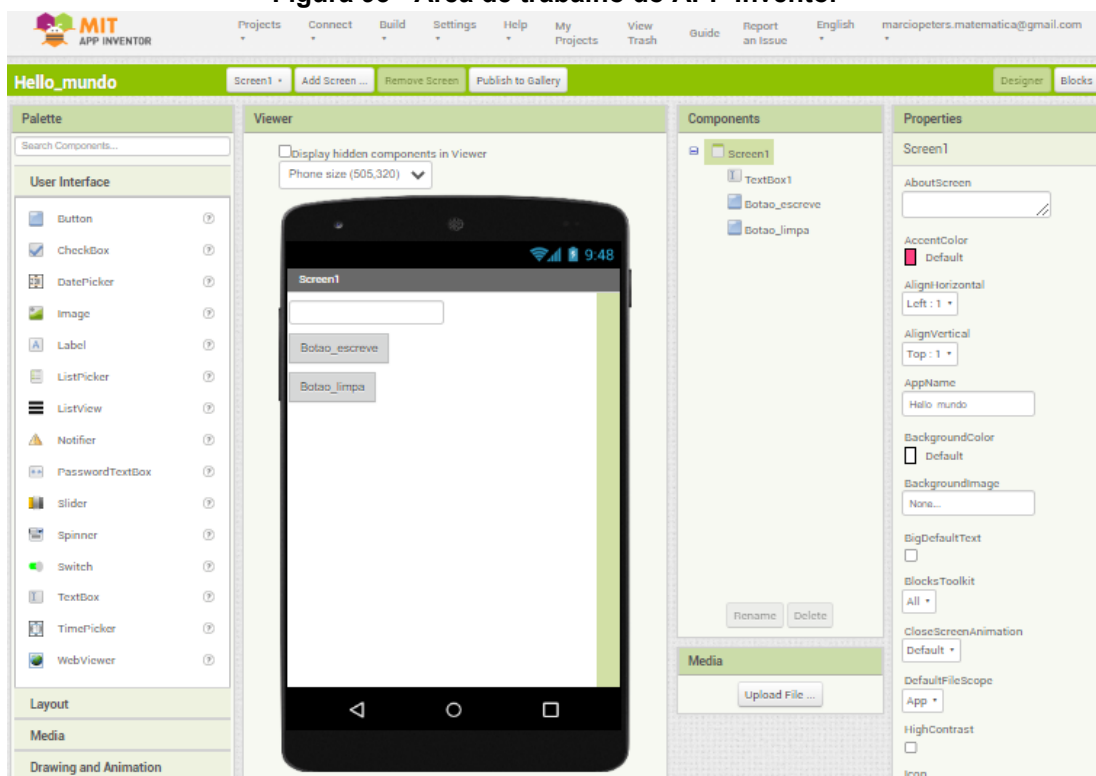
<sup>67</sup> Ambiente MIT APP Inventor. Disponível em: <<https://appinventor.mit.edu/>>. Acesso em 03/2022.

Explique aos estudantes que vão implementar inicialmente um aplicativo bem simples com objetivo de que o aplicativo irá possuir uma etiqueta (texto), uma caixa de texto para entrada de texto e dois botões: o primeiro botão escreve na caixa de texto a mensagem “Hello World”. E o segundo botão deleta essa mensagem.

Peça para os alunos arrastarem para “Screen 1” três componentes: um Textbox e dois botões. - Inicialmente os componentes têm nomes genéricos, peça para os alunos mudarem os nomes clicando no componente, e depois em Rename. Peça para renomear os botões, no campo “Texto” em “Propriedades”, para “Botao\_escreve” e “Botao\_limpa”.

Observe que, sempre iremos primeiro criar o nosso designer e em seguida a programação, ou seja, os comandos ou retorno daquilo que desejamos que o nosso aplicativo nos ofereça.

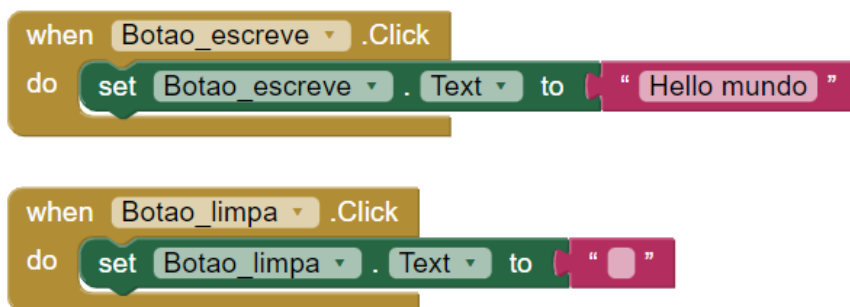
Figura 93 - Área de trabalho do APP Inventor



Fonte: Autoria própria, App Inventor

Em seguida, passe para o modo de programação clicando em “Blocks” ou “Blocos” e implemente o código abaixo, Figura 94.

**Figura 94 - Blocos de códigos do APP Inventor**



Fonte: Autoria própria, App Inventor

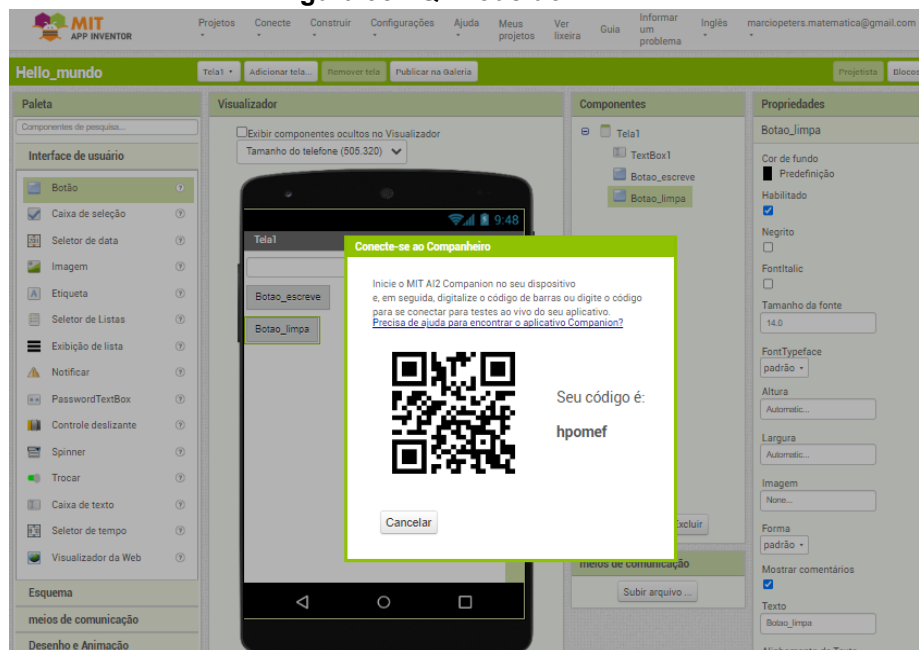
Agora, testamos o aplicativo com o emulador.

Caso tenha dúvidas como realizar os testes veja as instruções no link:

<http://appinventor.mit.edu/explore/ai2/setup>

Primeiro escolha uma das opções de testes indicados. Eu escolhi a opção de realizar teste com a instalação do MIT AI2 Companion no android. Com o aplicativo instalado no celular, ir à tela do App inventor, selecionar a opção Connect e, em seguida, a opção Companheiro de IA, e assim que abrir o código QR, Figura 95, basta ler pelo leitor do MIT AI2 Companion apontando a câmera para o código para realizar a conexão. Depois é só testar os resultados dessa oficina didática<sup>68</sup>!

**Figura 95 - Qr Code do MIT AI2**



Fonte: Autoria própria, App Inventor

<sup>68</sup> Fonte: Oficina didática adaptada de: <<https://naescola.codeiot.org.br/wp-content/uploads/2021/08/Curso-5-Semana-Implementando-o-primeiro-aplicativo.pdf>>. Acesso em 09/2022.

### 6.1.5.2 Oficina didática - Aplicativo de mensagens via Bluetooth

Criar aplicativos tem se tornado uma tarefa “fácil”, não necessariamente é preciso ser expert em programação, pois a utilização de ferramentas como o APP Inventor e outros, tem aberto diversas possibilidades:

“A criação de aplicativos de forma gráfica e interativa não necessariamente implica na utilização de algoritmos computacionais. É possível criar aplicativos apenas ligando alguns poucos componentes com mínima interação algorítmica entre eles.” (OLIVEIRA; PEREIRA; TEIXEIRA, 2021, p. 03).

O objetivo dessa oficina didática é, a partir dos conhecimentos prévios adquiridos na oficina didática - APP Inventor o estudante consiga desenvolver um aplicativo que permita troca de mensagens instantâneas entre dois dispositivos Android, explorar o Bluetooth, Figura 96, enquanto recurso de comunicação disponível para celulares e entender melhor assim como refletir sobre o processo de comunicação entre dispositivos.

Inicialmente garanta que os computadores utilizados já contêm o aiStarter instalado, e que já exista uma conta no App Inventor disponível para utilização.

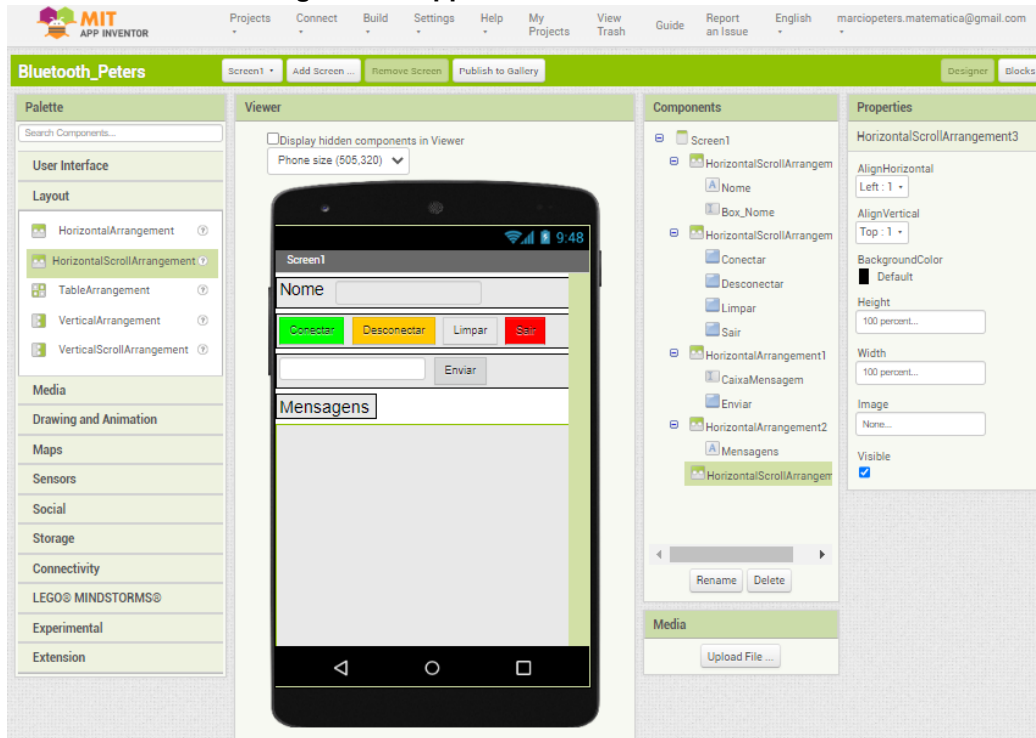
Para melhor condução das atividades divida a turma em duplas ou trios garantido assim que todos possam acompanhar o desenvolvimento das atividades.

Nesse momento, é importante alguns questionamentos:

- Como que os aplicativos como Messenger e Whatsapp funcionam?
- Em que situações cotidianas usamos Bluetooth e como ele funciona?
- O que um aplicativo de troca de mensagens precisa ter e que operações precisa fazer?

Após esse momento de reflexão já podemos iniciar acessando o site do App Inventor no link <http://appinventor.mit.edu/explore/>. Já na página clicar em “Create apps” ou “Crie aplicativos”, depois em “Start New Project” ou “Iniciar novo projeto”, e daí nomear o projeto (nome pode ser escolhido como queira ou ainda, colocar “Bluetooth\_Nome do estudante”. A partir daí, inclua os componentes que desejar que apareçam no interface do aplicativo que está sendo construído para a implementação do app, por exemplo, aqui temos botões, caixa de texto e ainda componentes invisíveis.

**Figura 96 - App comandos via Bluetooth**



Fonte: Aatoria própria.

Todo iniciante no APP Inventor encontra dificuldades básicas de como organizar uma tela. Por padrão, no App Inventor os componentes que incluímos são colocados um debaixo do outro. Mas, em algumas ocasiões, podemos optar por outro tipo de organização. Para isso, o App Inventor dispõe de uma seção chamada “ScreenArrangement” (Organização em tela) que dispõe de três componentes para criar desenhos com a organização que desejarmos, isto é, na paleta de ferramentas do APP Inventor temos o “Layout” que contém várias opções de formatos para enfeitar o seu designer do seu aplicativo.

Agora é hora da programação do aplicativo, para isso o próximo passo é clicar em Blocos e implementar os comandos de cada um dos elementos do seu aplicativo.

Para auxiliar, caso necessário, acesse o tutorial disponível em <http://appinventor.mit.edu/explore/ai2/setup>, pois, assim, você pode compreender melhor como configurar o APP Inventor para testar seu aplicativo enquanto você cria.

Auxiliar os estudantes quanto às funções dos diferentes blocos, assim como, quais deles serão necessários para implementação do app e também a forma de organização do layout é muito importante para que se tenha um bom resultado.

Na fase de testes com o emulador, o programa aiStarter, deve-se, após os passos de montagem dos elementos ou componentes, seguir com a programação lógica, ou “Blocos”. Encoraje os estudantes a fazerem sozinhos após a primeira oficina didática, e peça aos estudantes que conseguirem, que auxiliem aqueles que continuam na tentativa, essa fase de aprendizagem e troca é muito importante para o crescimento do grupo.

Já na fase de reflexão abordar os seguintes questionamentos, entre outros:

- Quais outras informações podemos transmitir por meio de um aplicativo?
- Qual a importância de poder transmitir informações?
- Qual foi a parte mais difícil da atividade? E a mais fácil?

Deixe claro que o objetivo não é que os estudantes criem um aplicativo perfeito, mas que explorem as possibilidades aprendidas até o momento, utilizem a criatividade, e que cada um possa através dessas atividades fazer um diagnóstico da recepção e entendimento a respeito dos conteúdos.

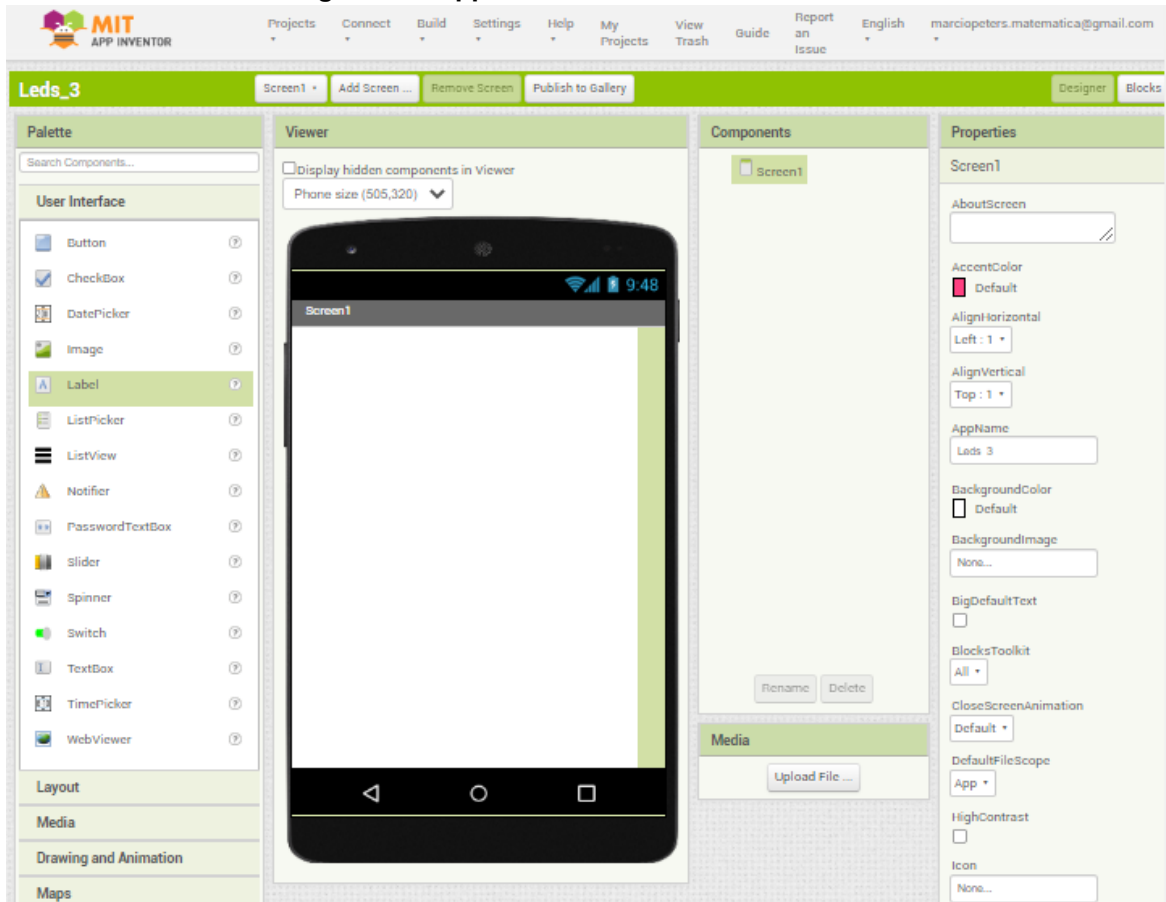
Fonte: Oficina didática adaptada de: <<https://naescola.codeiot.org.br/wp-content/uploads/2021/10/Curso-5-Semana-4-Bluetooth.pdf>>. Acesso em 03/2022.

#### 6.1.5.3 Oficina didática - Aplicativo para controlar LEDs via Bluetooth

O objetivo dessa oficina didática é a montagem de um aplicativo que controla 3 Leds, Figura 97, e a partir daí, desenvolver as habilidades básicas para montarmos outros aplicativos que irão auxiliar na construção de projetos mais complexos considerando a necessidade e complexidade de cada um dos projetos que tiverem em mente.

Para iniciarmos, faça login no ambiente do App Inventor (<http://appinventor.mit.edu/>) com seu e-mail Google. Em seguida, crie novo projeto em “projetos” depois “novo projeto”, nomeie seu novo projeto, eu coloquei “Led\_3”, e assim irá direcioná-lo para a janela de trabalho.

Figura 97 - App controlando LED via Bluetooth



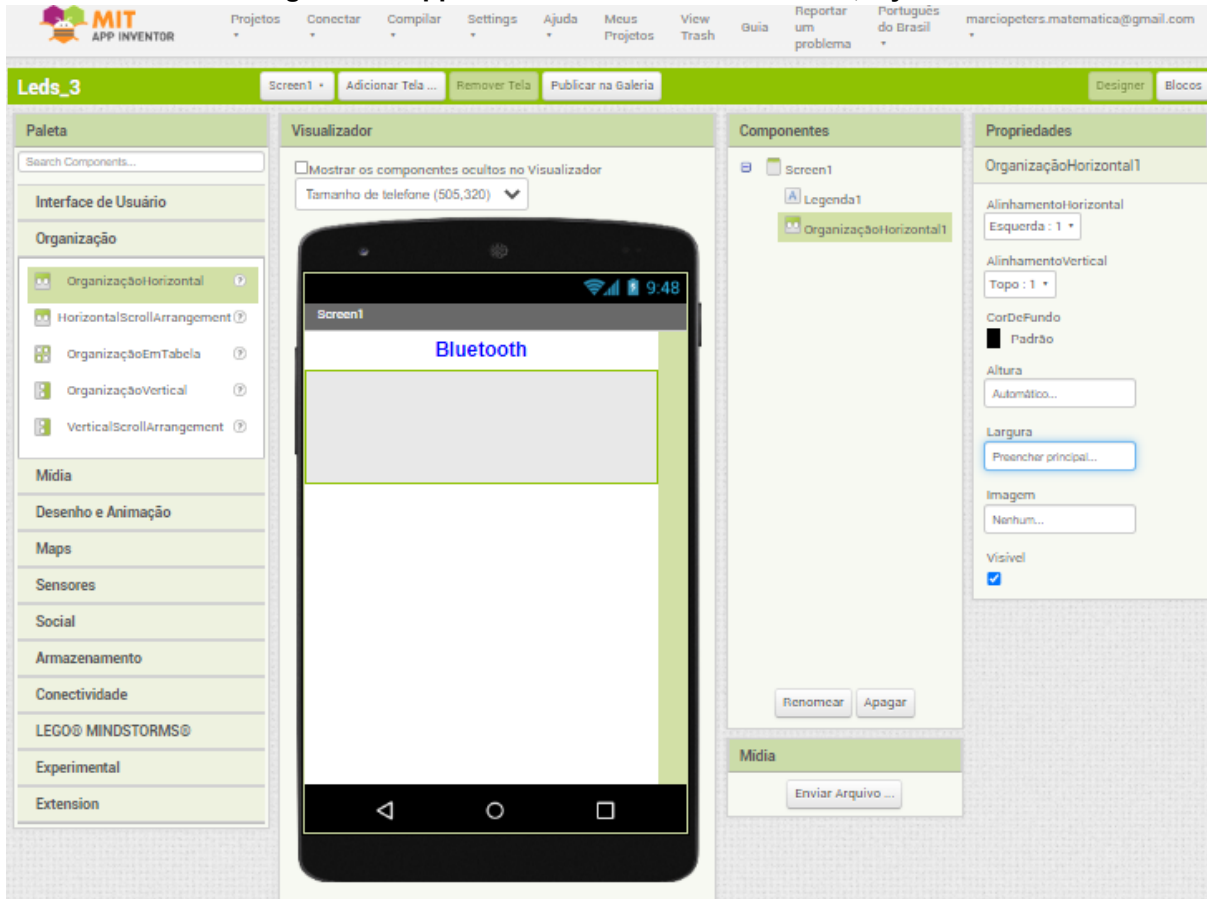
Fonte: Autoria própria.

**Observação:** Prefira trabalhar com o aplicativo na linguagem que domina (língua portuguesa), e para isto, caso queira, basta alterar o idioma na barra superior próximo ao seu nome de usuário ou e-mail.

Arraste para a área de trabalho uma “Legenda” renomeie a caixa texto “Bluetooth” e para uma melhor visualização altere a fonte para 20 com alinhamento centralizado. Em seguida na caixa largura selecione “Preencher principal” e clique “Ok”, repare que a caixa de texto irá ocupar a tela toda na horizontal, selecione também a fonte negrito e altere a cor de texto, coloquei azul, mas fica a sua escolha, ainda na parte inferior de “Propriedades”.

Agora, no recurso “Layout”, Figura 98, ou traduzido “Organização” escolha “OrganizaçãoHorizontal” arraste para a área de trabalho e configure a propriedade para “preencher principal” selecionando na caixa “largura”, note que a caixa irá completar toda a faixa do aplicativo.

Figura 98 - App controlando LED via Bluetooth, layout



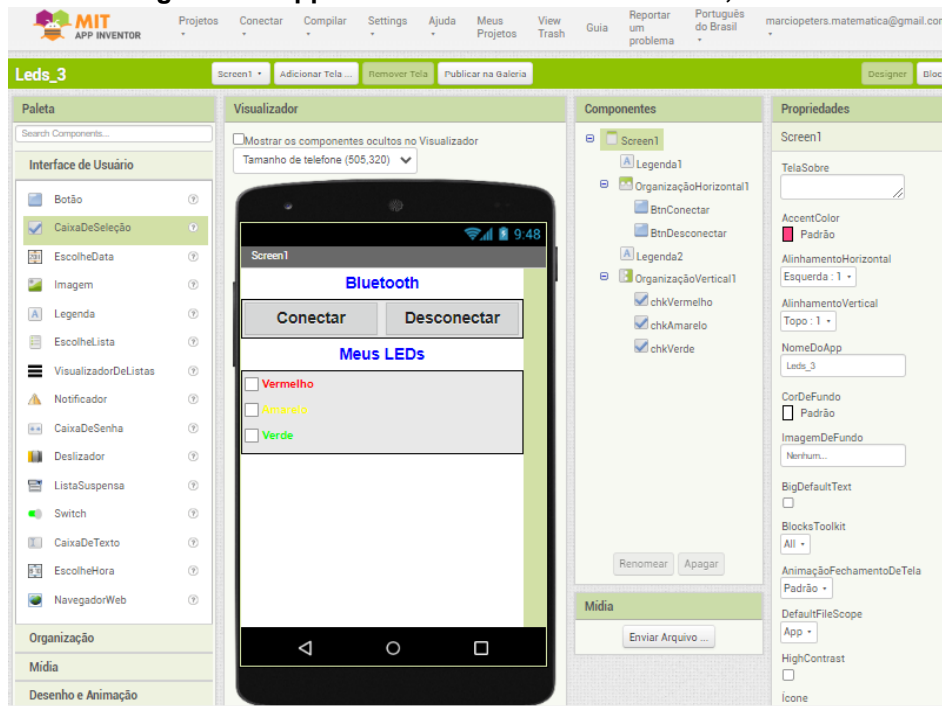
Fonte: A autoria própria.

Feito isso, inclua dois botões, Figura 99, dentro dessa caixa, para isso vá em “Interface do Usuário” e arraste “Botão” duas vezes, em seguida nomeie cada um deles. No caso foi colocado “BtnConectar” com texto “Conectar” em negrito e outro “BtnDesconectar” com texto “Desconectar” em negrito, alinhado ao centro para ambos. Em seguida arraste mais uma “Legenda” abaixo dos botões, renomeie na caixa texto para “Meus LEDs” com as propriedades “fonte 20”, negrito e centralizado, na caixa largura escolha “Preencher principal”.

Para podermos incluir os botões de checagens de forma organizada, vá em “Organização” arraste “OrganizaçãoVertical” e escolha as propriedades “largura”, “Preencher principal” para ocupar a tela toda, em seguida arraste 3 “Caixa de seleção” para essa caixa. A primeira renomeie para “chkVermelho” com texto “Vermelho” fonte 20, negrito cor vermelha e centralizado, na caixa largura escolha “Preencher principal”. Repita esse procedimento para as caixas de seleção “Amarelo” e “Verde”.



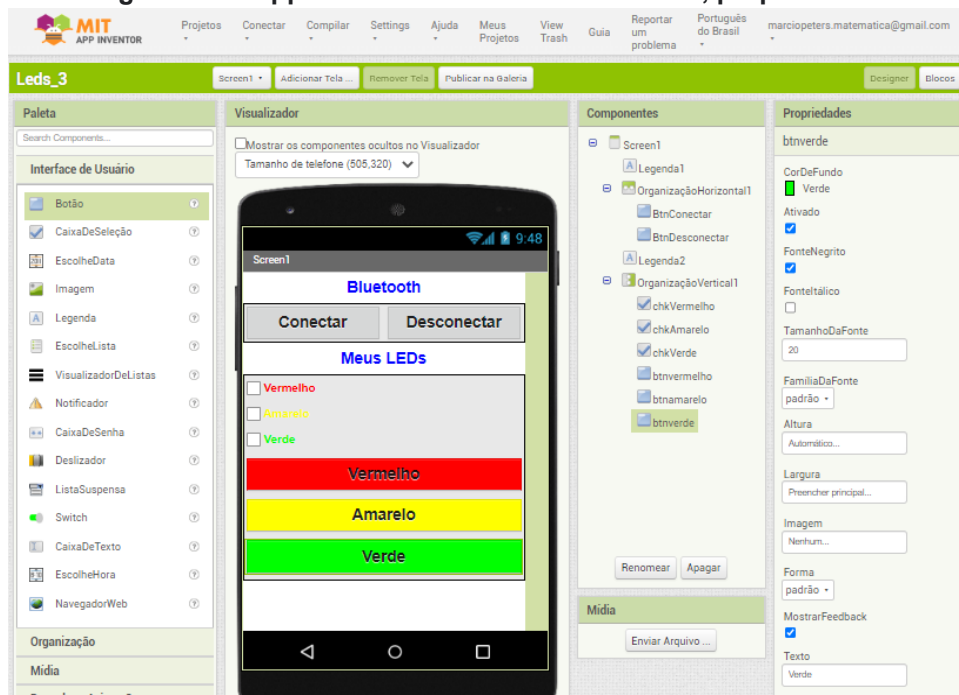
Figura 99 - App controlando LED via Bluetooth, botões



Fonte: Autoria própria.

Agora, inclua ainda 3 botões, para o primeiro renomeie “btnvermelho” com as propriedades, Figura 100, cor de fundo vermelha, fonte 20, negrito e centralizado, na caixa largura escolha “Preencher principal” e na caixa “Texto” escreva “Vermelho”. Repita esse procedimento para os botões “Amarelo” e “Verde”.

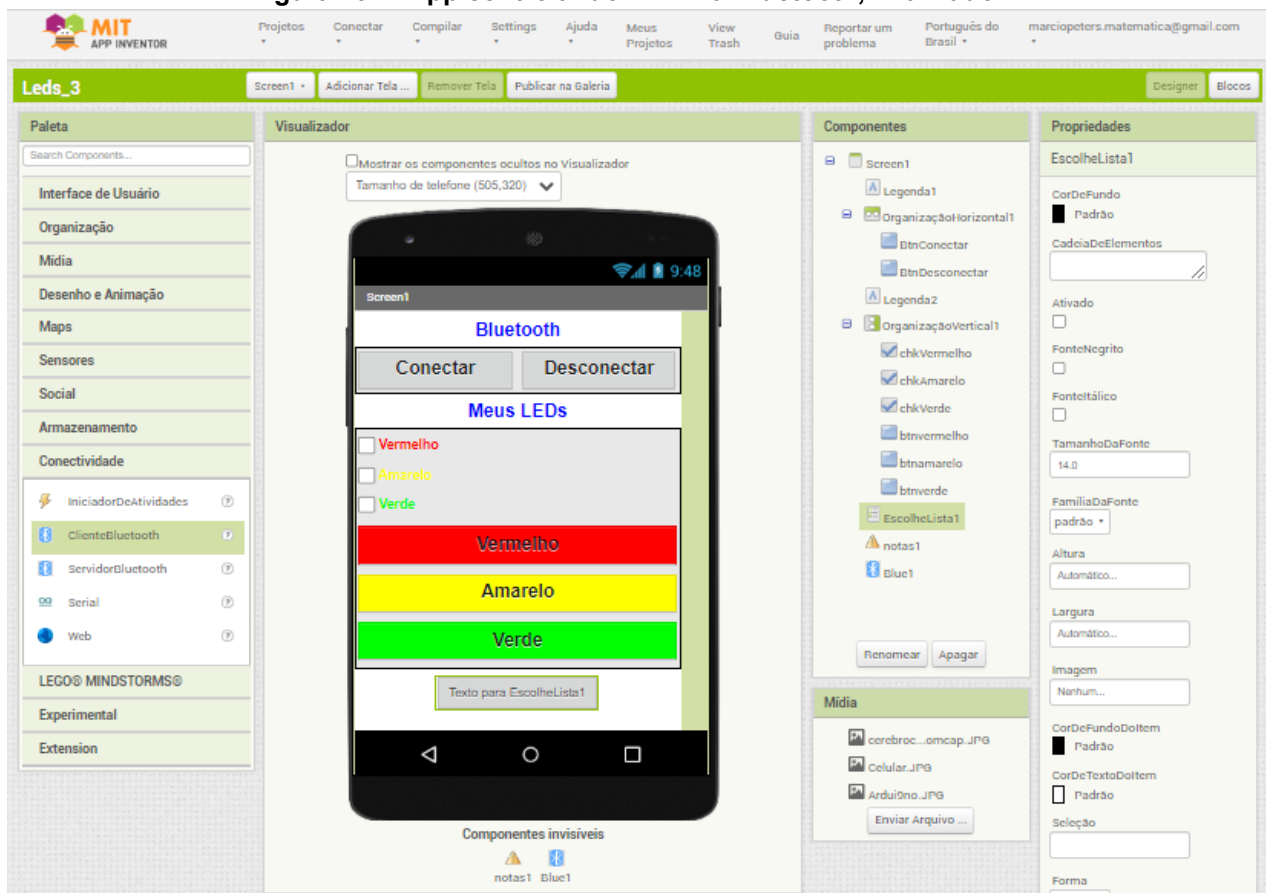
Figura 100 - App controlando LED via Bluetooth, propriedades



Fonte: Autoria própria.

Nossa interface do aplicativo, Figura 101, já está quase pronta para darmos início aos “Blocos” de comando, Figura 102, isto é, a programação de cada um desses elementos. Iremos incluir alguns componentes que ficarão ocultos dentro do aplicativo, mas que serão importantes para a nossa programação, que são eles “EscolheLista”, um “Notificador”, e um “ClienteBluetooth” na paleta “Conectividade”, arraste cada um destes da “Interface de Usuário” para a tela de trabalho. Renomeie o “Notificador” para “notas1” e o “ClienteBluetooth” para “Blue1”.

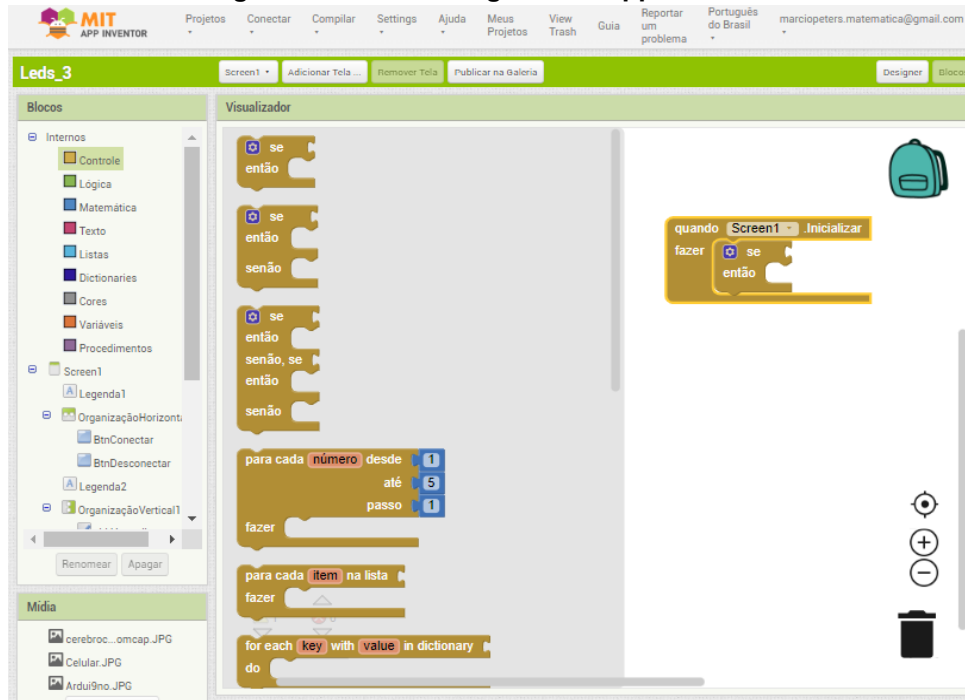
**Figura 101 - App controlando LED via Bluetooth, finalizado**



Fonte: Autoria própria.

Na parte lógica “Blocos”, selecione “Screen1”, arraste o bloco “quando screen1 inicializar, fazer” e em seguida “Controle” arraste o bloco “se, então”.

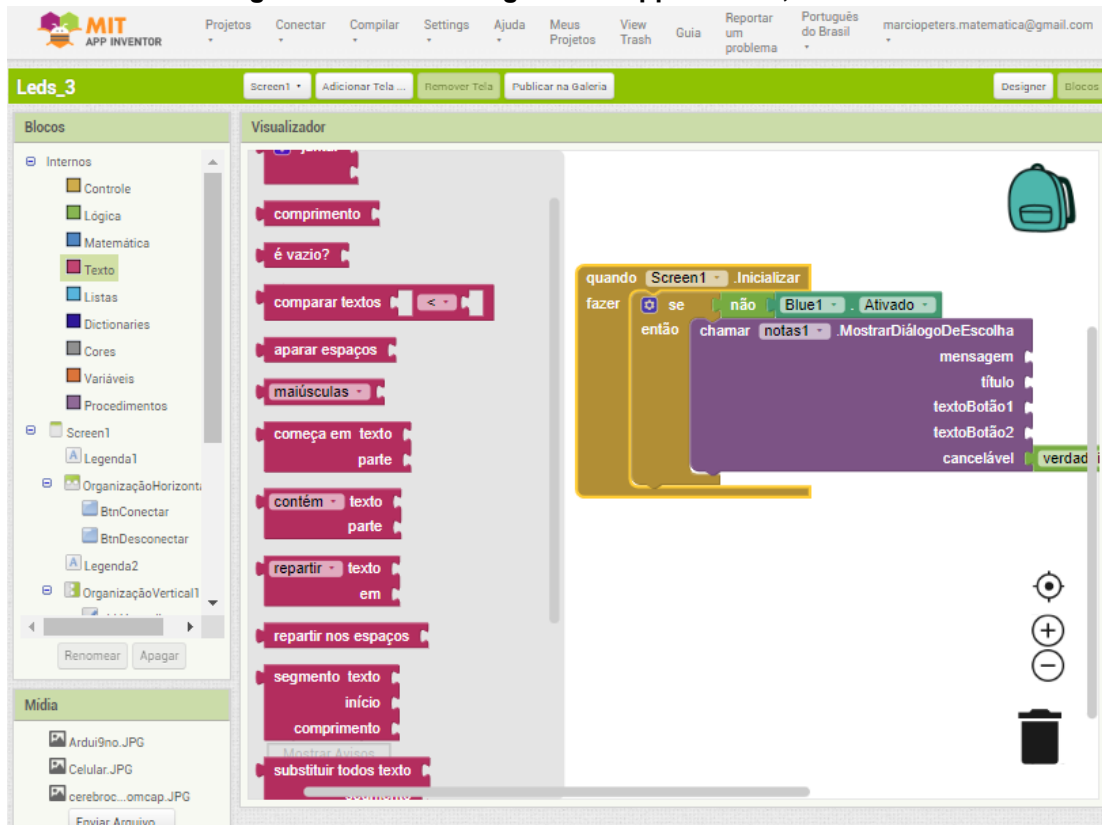
Figura 102 - Blocos lógicos do App Inventor



Fonte: Autoria própria.

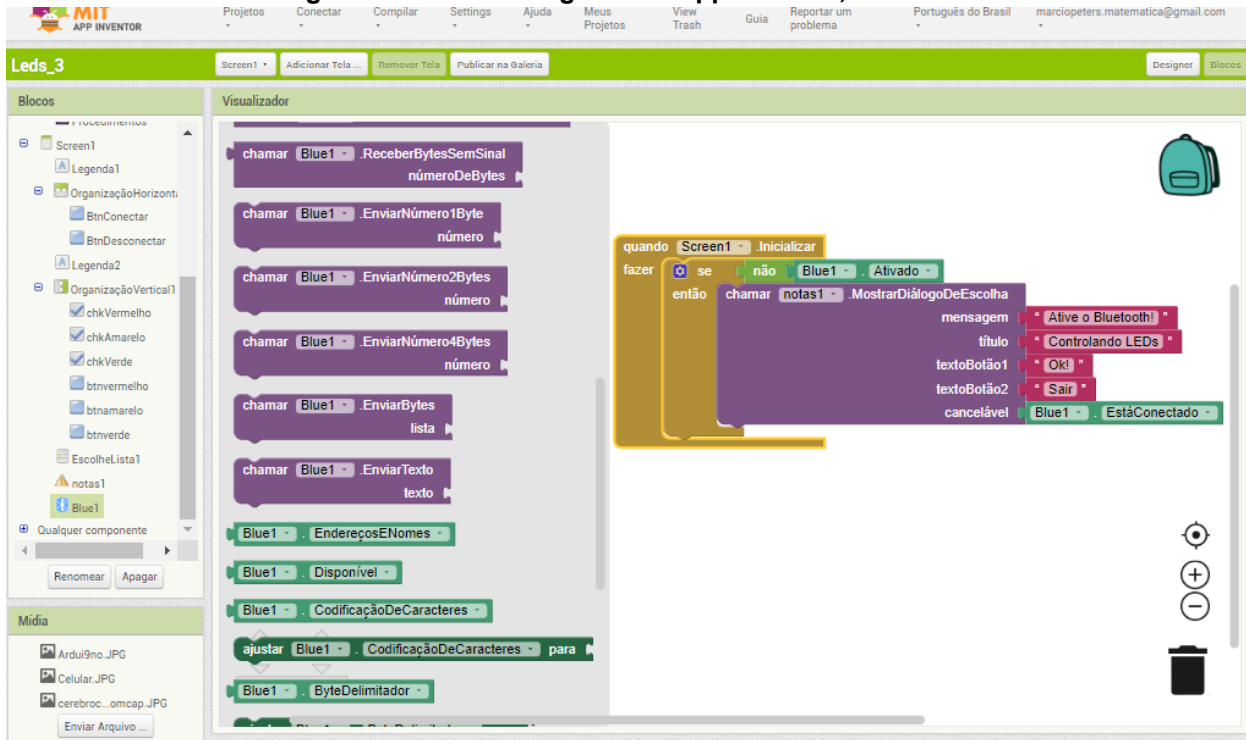
Na parte lógica, se o Bluetooth não estiver conectada teremos uma mensagem no “notas1”, Figura 103.

Figura 103 - Blocos lógicos do App Inventor, notas 1



Fonte: Autoria própria.

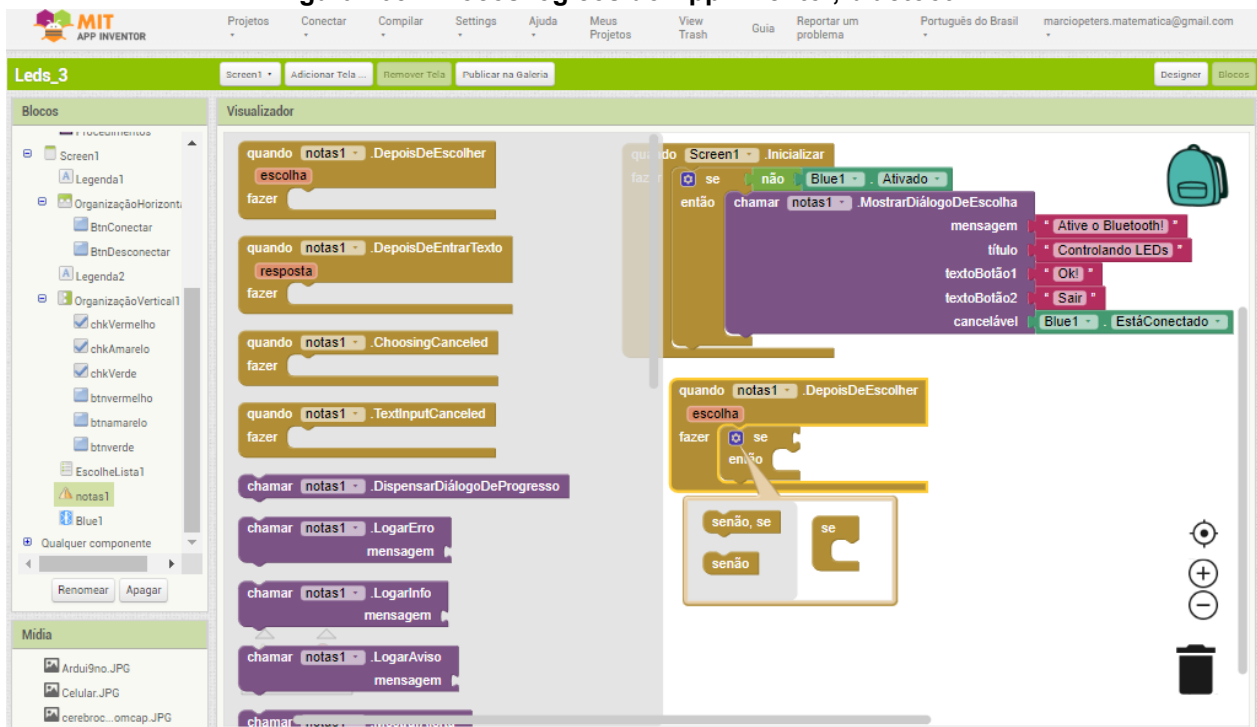
Figura 104 - Blocos lógicos do App Inventor, screen 1



Fonte: Autoria própria.

Após essa primeira parte da Screen1, iremos criar o bloco da parte de mensagens “notas1”, Figura 104.

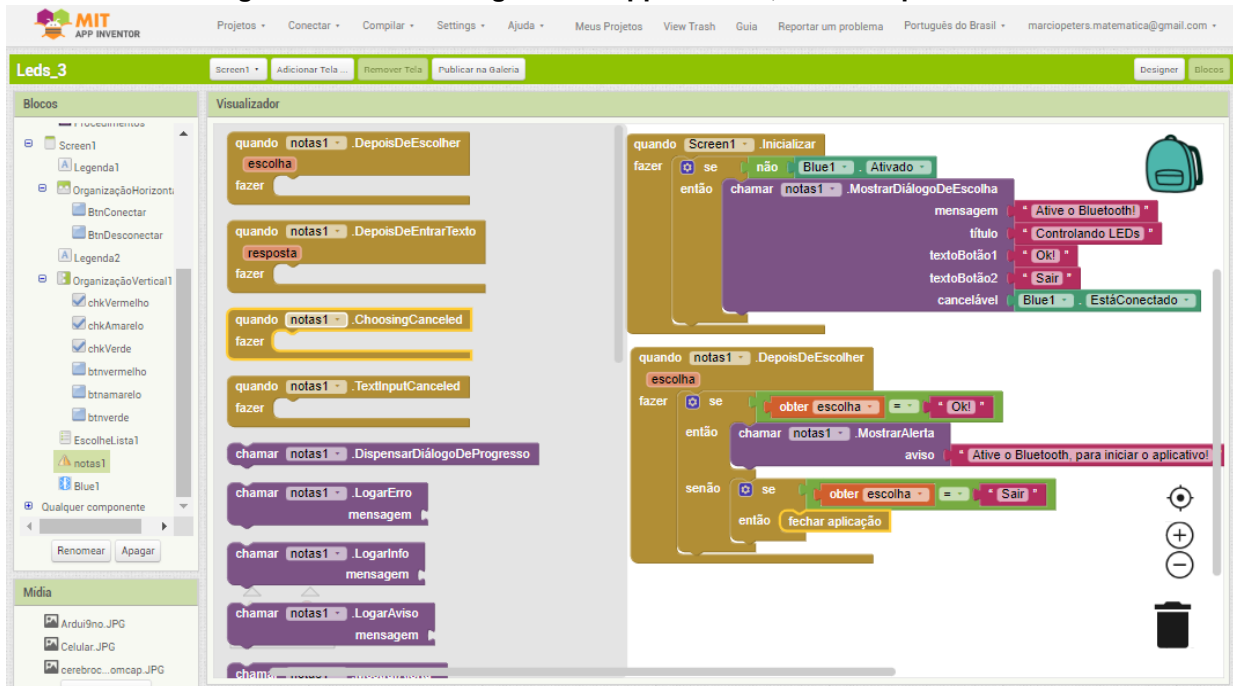
Figura 105 - Blocos lógicos do App Inventor, bluetooth



Fonte: Autoria própria.

Observe que durante a montagem dos blocos de comandos, Figura 105, alguns itens de controle ficam suspensos, Figura 106, e é preciso implementar durante a construção dos blocos lógicos, essa “expertise” o usuário irá adquirindo com tempo e prática.

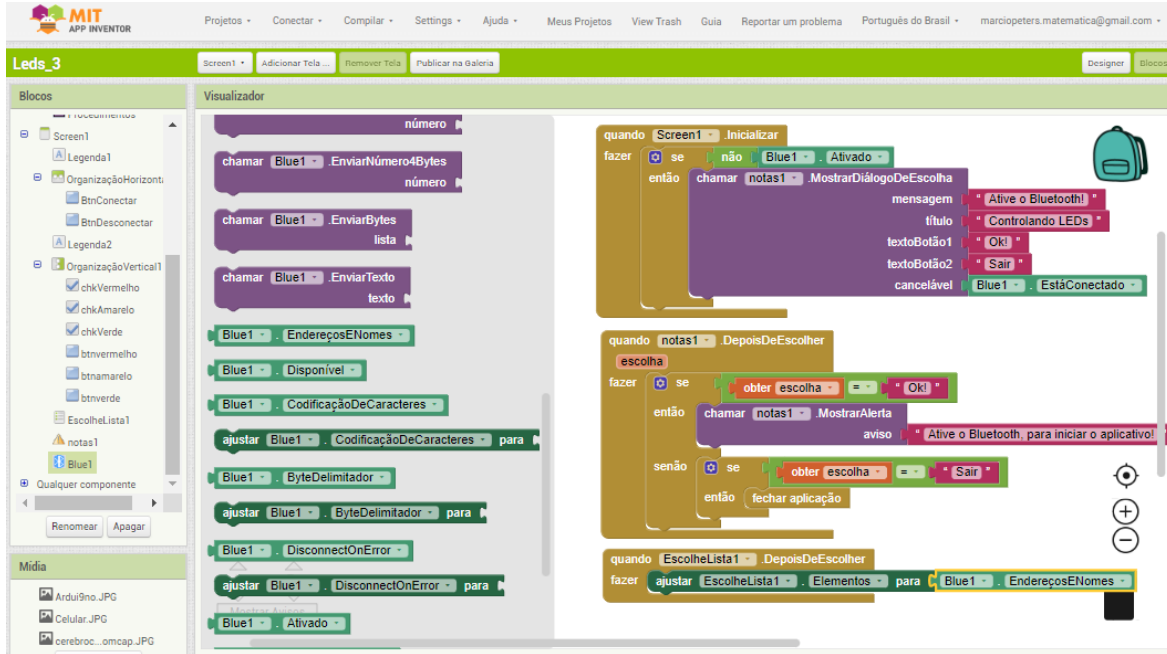
**Figura 106 - Blocos lógicos do App Inventor, itens suspensos**



Fonte: Autoria própria.

Nosso próximo passo é escrever o bloco lógico da nossa lista, para isso, selecione lista e implemente o código conforme a Figura 107, essa lógica é para encontrar os dispositivos Bluetooth próximos.

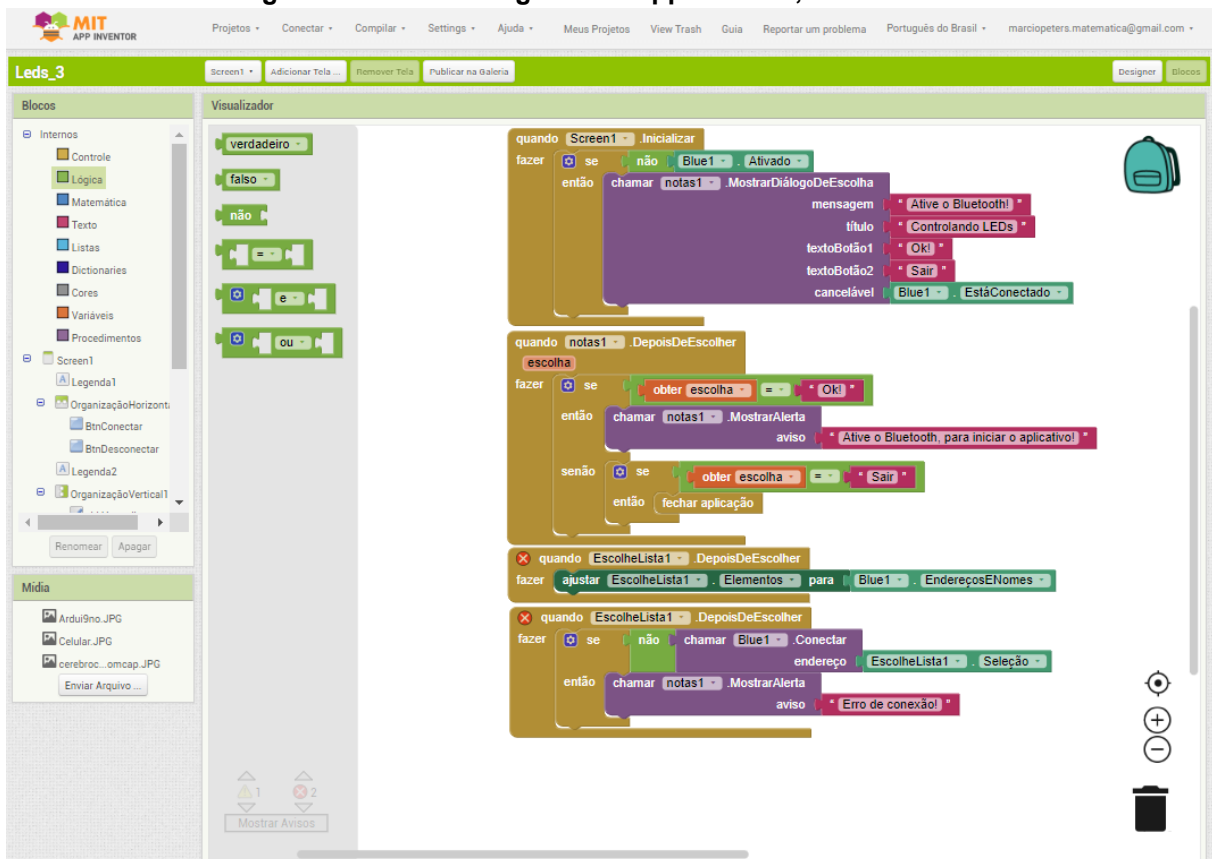
Figura 107 - Blocos lógicos do App Inventor, busca do dispositivo



Fonte: Autoria própria.

Continuando nossa implementação, clique “EscolheLista1”, Figura 108, para a montagem lógica desse bloco de comandos, veja na imagem abaixo os detalhes.

Figura 108 - Blocos lógicos do App Inventor, escolhe lista

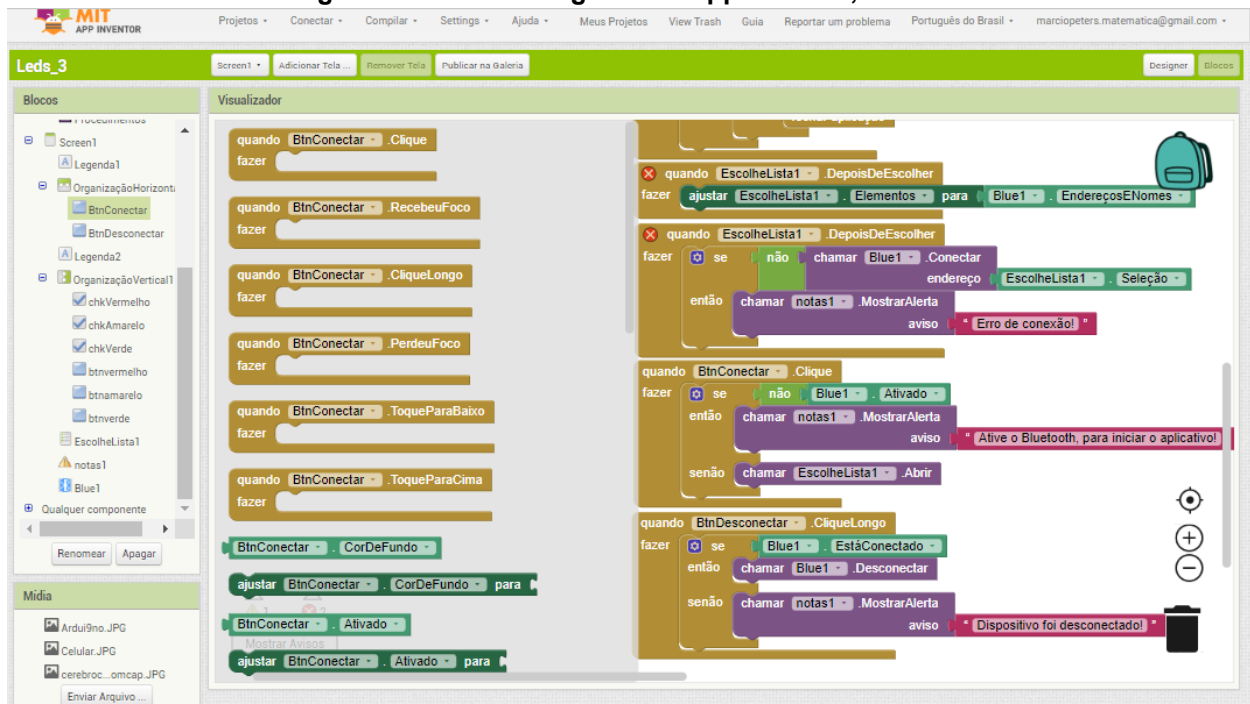


Fonte: Autoria própria.

Por fim iremos criar os blocos de comandos dos dois botões, Figura 109, “Conectar” e “Desconectar” que irão chamar essas funções anteriormente criadas.

Observação: Criado os blocos de comandos, a exemplo do bloco Bluetooth, eles são padrões para outros projetos e assim basta seguir a mesma linha de raciocínio e isso agiliza processos de construção.

**Figura 109 - Blocos lógicos do App Inventor, botões**

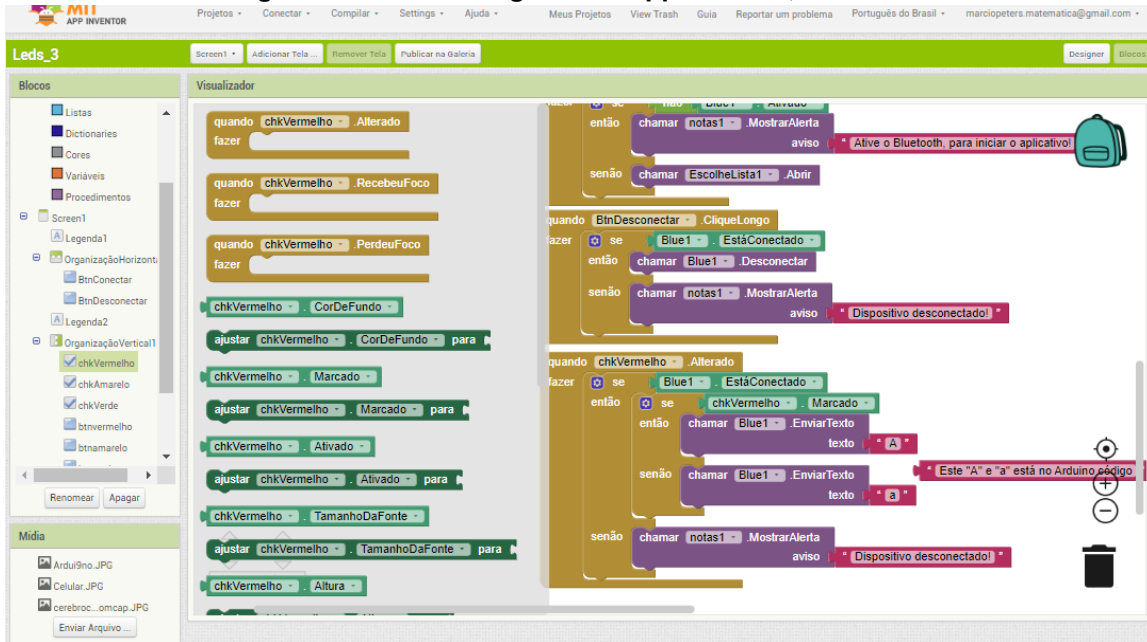


Fonte: Autoria própria.

Observe que essa lógica criada até aqui será sempre útil e padrão para todos os aplicativos que irão conectar via Bluetooth, então vale a pena salvar essa primeira parte para utilização em outros projetos que contém esse mesmo quesito.

Agora iremos para a parte de blocos que irão compor os checkbox, Figura 110, na parte de baixo do aplicativo. Crie o primeiro bloco, checkbox “vermelho”, e depois duplica para os outros dois “verde” e “amarelo”. Veja, Figura 111, o bloco já implementado.

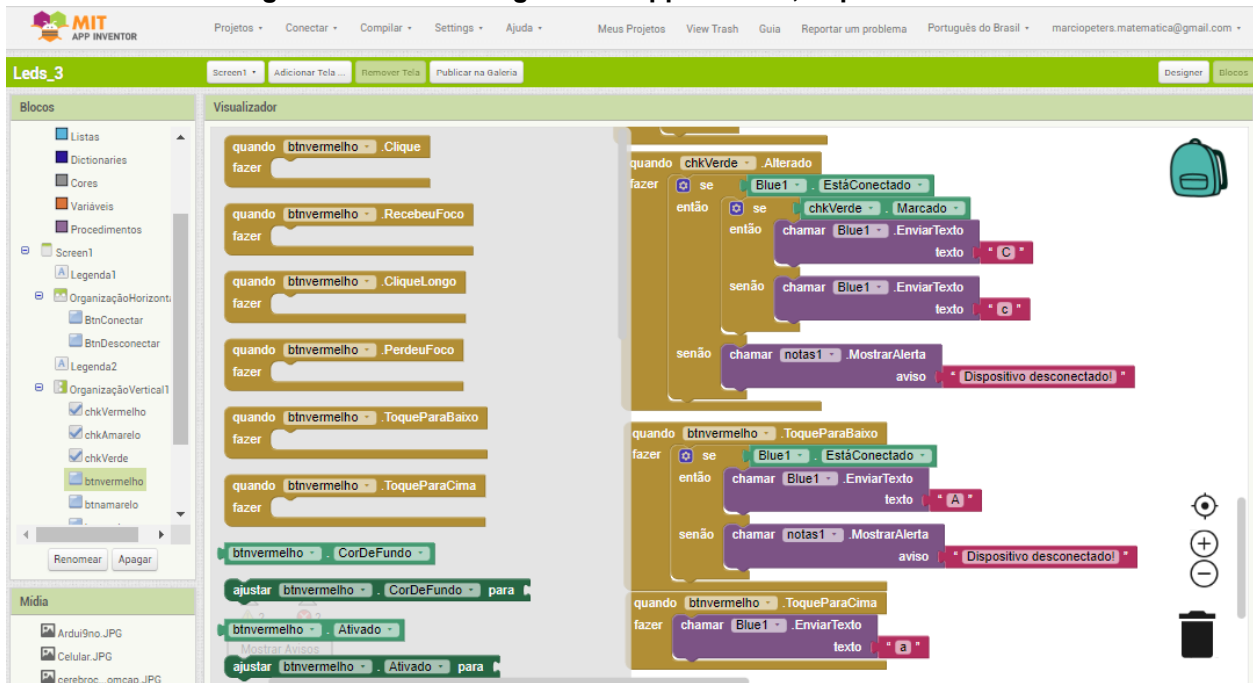
Figura 110 - Blocos lógicos do App Inventor, checkbox



Fonte: Autoria própria.

Criado os blocos dos três checkbox, só faltam os blocos dos 3 botões, vermelho, amarelo e verde, basta fazer um deles e depois duplicar para os demais.

Figura 111 - Blocos lógicos do App Inventor, duplicar botões



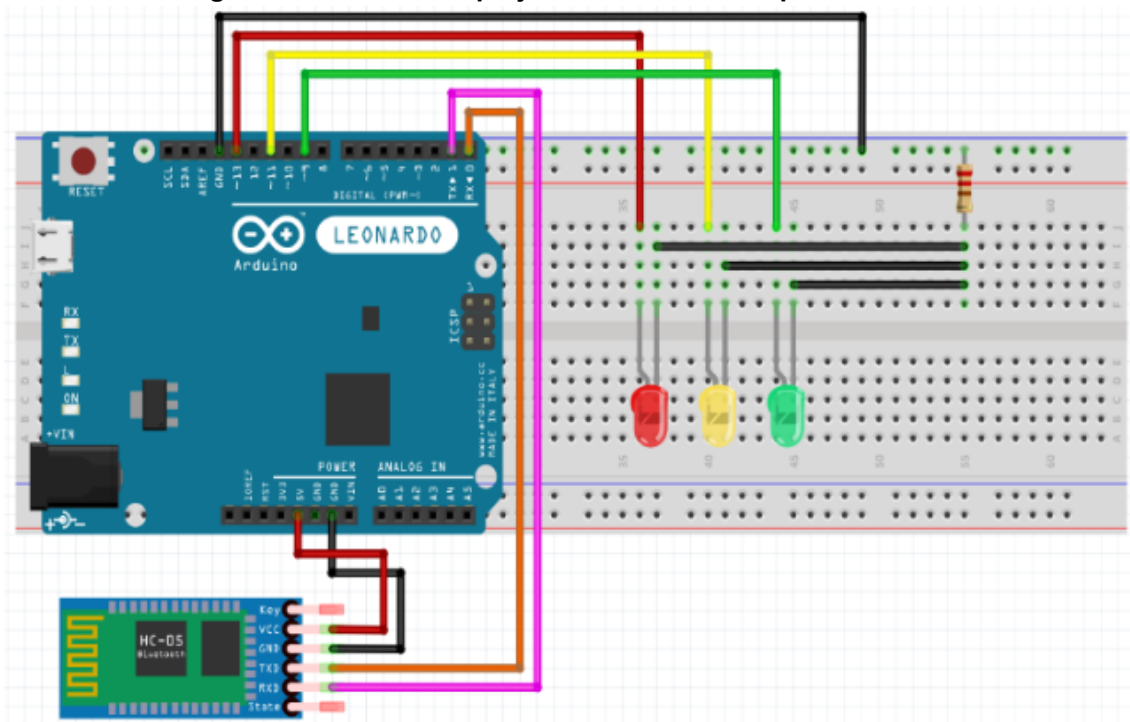
Fonte: Autoria própria.

Agora o próximo passo é testar o aplicativo no seu celular.



Esquema de ligação dos componentes, Figura 112.

Figura 112 - Sketch do projeto LED controlado por Bluetooth



Fonte: Site [sergiogracas.com](http://sergiogracas.com)<sup>69</sup>

Dando continuidade após a certificação das ligações corretas com os componentes ao projeto, é hora de carregar o código ao controlador Arduino, segue:

```
int ledverde = 9;
int ledamarelo = 11;
int ledvermelho = 13;
void setup() {
  //informando os pinos que são de saída//
  pinMode(ledverde, OUTPUT);
  pinMode(ledamarelo, OUTPUT);
  pinMode(ledvermelho, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  //Criando uma variável do tipo caractere
  char z;
  //variável 'c' recebe o valor da porta Serial
  z =Serial.read();
```

<sup>69</sup> Disponível em: <[http://sergiogracas.com/html5/flisol\\_2015/#13](http://sergiogracas.com/html5/flisol_2015/#13)>. Acesso em 03/2022.

```

// Checagem/teste de valores, se 'C' recebido acender led verde
if (z=='C'); digitalWrite (ledverde, HIGH);
// Checagem/teste de valores, se 'c' recebido apagar led verde
if (z=='c'); digitalWrite (ledverde, LOW);

// Checagem/teste de valores, se 'B' recebido acender led amarelo
if (z=='B'); digitalWrite (ledamarelo, HIGH);
// Checagem/teste de valores, se 'b' recebido apagar led amarelo
if (z=='b'); digitalWrite (ledamarelo, LOW);
// Checagem/teste de valores, se 'A' recebido acender led vermelho
if (z=='A'); digitalWrite (ledvermelho, HIGH);
// Checagem/teste de valores, se 'a' recebido apagar led vermelho
if (z=='a'); digitalWrite (ledvermelho, LOW);
delay(500);
}

/* Fonte: Adaptado de
// https://www.youtube.com/watch?v=blvkJBACGY0&t=184s */

```

Essa integração do Arduino, Bluetooth, Android com AppInventor é bastante trabalhosa, mas ao final vale a pena, pois, a complexidade nos possibilita o desenvolvimento de habilidades e competências para nosso cotidiano e aplicações em projetos variados posteriormente.

#### 6.1.5.4 Oficina didática - Aplicativo para controlar motor DC via Bluetooth

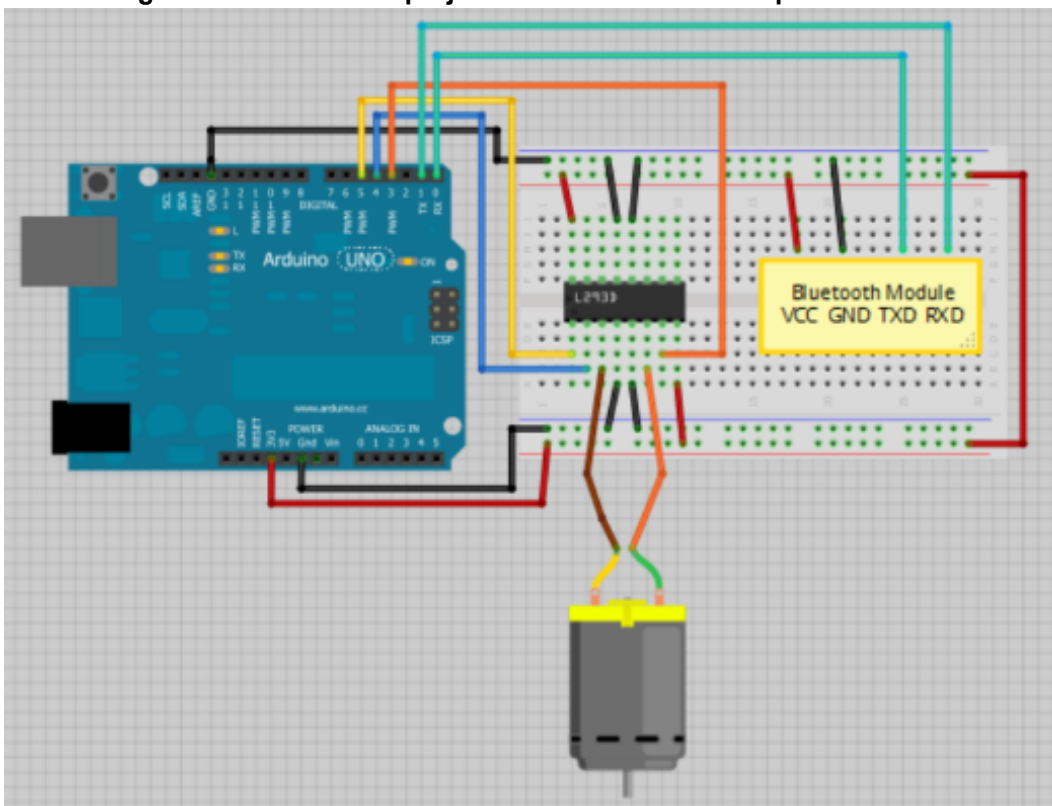
Essa oficina didática tem como objetivo apresentar uma forma de comunicação entre seu smartphone e o Arduino e para isso, iremos conhecer um app que faça a conexão entre motores DC via bluetooth com um aplicativo Android. Ainda, com essa oficina didática é possível entender um pouco mais sobre motores DC, interface do Arduino com o seu smartphone, Bluetooth e o L293D IC (é um circuito integrado dedicado ao controle de pequenos motores DC), Figura 113.

Caso você não tenha o IC L293 você pode fazer o mesmo circuito usando a ponte H (um circuito para variar a corrente em uma determinada carga), de qualquer forma é recomendável que leia mais sobre isso e o datasheet do IC.

Material necessário:

1. Arduino UNO;
2. um Módulo Bluetooth (por exemplo: HC-05 ou 06);
3. um Smartphone (qualquer Android funcionará);
4. Aplicativo BlueTerm;
5. um L293D CI;
6. um motor DC;
7. uma protoboard;
8. cabos de ligação (Jumpers);
9. um notebook/desktop para cada aluno ou grupo de alunos.

Figura 113 - Sketch do projeto Motor DC controlado por Bluetooth



Fonte: Randomnerdtutorials<sup>70</sup>

Observações:

Remova os cabos RX e TX quando estiver carregando o esboço para o seu Arduino.

Às vezes as pessoas conectam o TX do módulo bluetooth ao TX do Arduino... isso está errado e não vai funcionar. Certifique-se de conectá-los corretamente, o TX no RX e o RX no TX.

<sup>70</sup> Adaptado de Randomnerdtutorials: <<https://randomnerdtutorials.com/arduino-control-dc-motor-via-bluetooth/>>. Acesso em 03/2022.

**Nota:**

Se o Módulo Bluetooth HC-05 solicitar uma senha, é '1234' ou ainda "0000".

Código do projeto:

```
/**Controle motor DC com Smartphone via bluetooth
 * Criado por Rui Santos, http://randomnerdtutorials.com */
int motorPin1 = 3; // pino 2 no L293D IC
int motorPin2 = 4; // pino 7 no L293D IC
int enablePin = 5; // pino 1 no L293D IC
int state;
int flag=0; //garante que o serial seja impresso apenas uma vez no
estado
void setup() {
    // Define os pinos como saída:
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(enablePin, OUTPUT);
    digitalWrite(enablePin, HIGH); // define habilitar pino alto
para que o motor possa ligar:
    Serial.begin(9600); // inicializa a comunicação serial em 9600
bits por segundo:
}
void loop() {
    //se alguma data for enviada, lê e salva no estado
    if(Serial.available() > 0) {
        state = Serial.read();
        flag=0;
    }
    // Se o estado for '0' o motor DC irá desligar
    if (state == '0') {
        digitalWrite(motorPin1, LOW); // Defina pino 2 no L293D
low
        digitalWrite(motorPin2, LOW); // Defina pino 7 no L293D
low
        if(flag == 0){
            Serial.println("Motor: off");
            flag=1;
        }
    }
    // Se o estado for '1' o motor DC irá ligar, virando a
esquerda
    else if (state == '1') {
```

```

        digitalWrite(motorPin1, LOW); // Defina pino 2 no L293D
desligado
        digitalWrite(motorPin2, HIGH); // Defina pino 7 no L293D
ligado
        if(flag == 0){
            Serial.println("Motor: right");
            flag=1;
        }
    }
    // Se o estado for '2' o motor virará para a esquerda
    else if (state == '2') {
        digitalWrite(motorPin1, HIGH); // Defina pino 2 no L293D
ligado
        digitalWrite(motorPin2, LOW); // Defina pino 7 no L293D
desligado
        if(flag == 0){
            Serial.println("Motor: left");
            flag=1;
        }
    }
}
}

```

Para a comunicação do android com o módulo bluetooth use o aplicativo BlueTerm, é totalmente gratuito, então você só precisa ir à “Play store” e fazer o download. Então você só precisa conectar seu smartphone com o módulo bluetooth. Lembre-se de remover os cabos TX e RX. Configure apenas 3 comandos para controlar o motor DC:

- '0' – Desliga o motor DC.
- '1' – Motor DC gira para a direita.
- '2' – Motor DC gira para a esquerda.

Fonte: Oficina adaptada de randomnerdtutoriais: Disponível em: <<https://randomnerdtutoriais.com/arduino-control-dc-motor-via-bluetooth/>>. Acesso 11/2022.

## 6.2 Computação desplugada

As atividades de computação desplugadas são aquelas que não dependem de equipamentos eletrônicos, tais como computadores, celulares ou tablets durante a execução das tarefas. Esta abordagem de aprendizagem de programação se concentra em atividades práticas e lúdicas que ajudam a ensinar os conceitos básicos de programação, como sequência, lógica e estrutura de dados.

### 6.2.1 Circuito de aprendizagem dinâmicas e atividades desplugadas

#### 6.2.1.1 Oficina didática - Controle remoto

O objetivo dessa oficina didática é exercitar os pilares de abstração, decomposição e algoritmos através da utilização de botões de eventos. A atividade deve ser realizada em dupla, sendo que uma das crianças fica com o controle e a outra executa as ações que constam no cartão de programação.

Material necessário:

- Uma folha contendo o controle remoto e cartões de programas para serem recortados, observe que a linha pontilhada indica um corte, mas, deve ser de forma a encaixar o cartão no orifício.

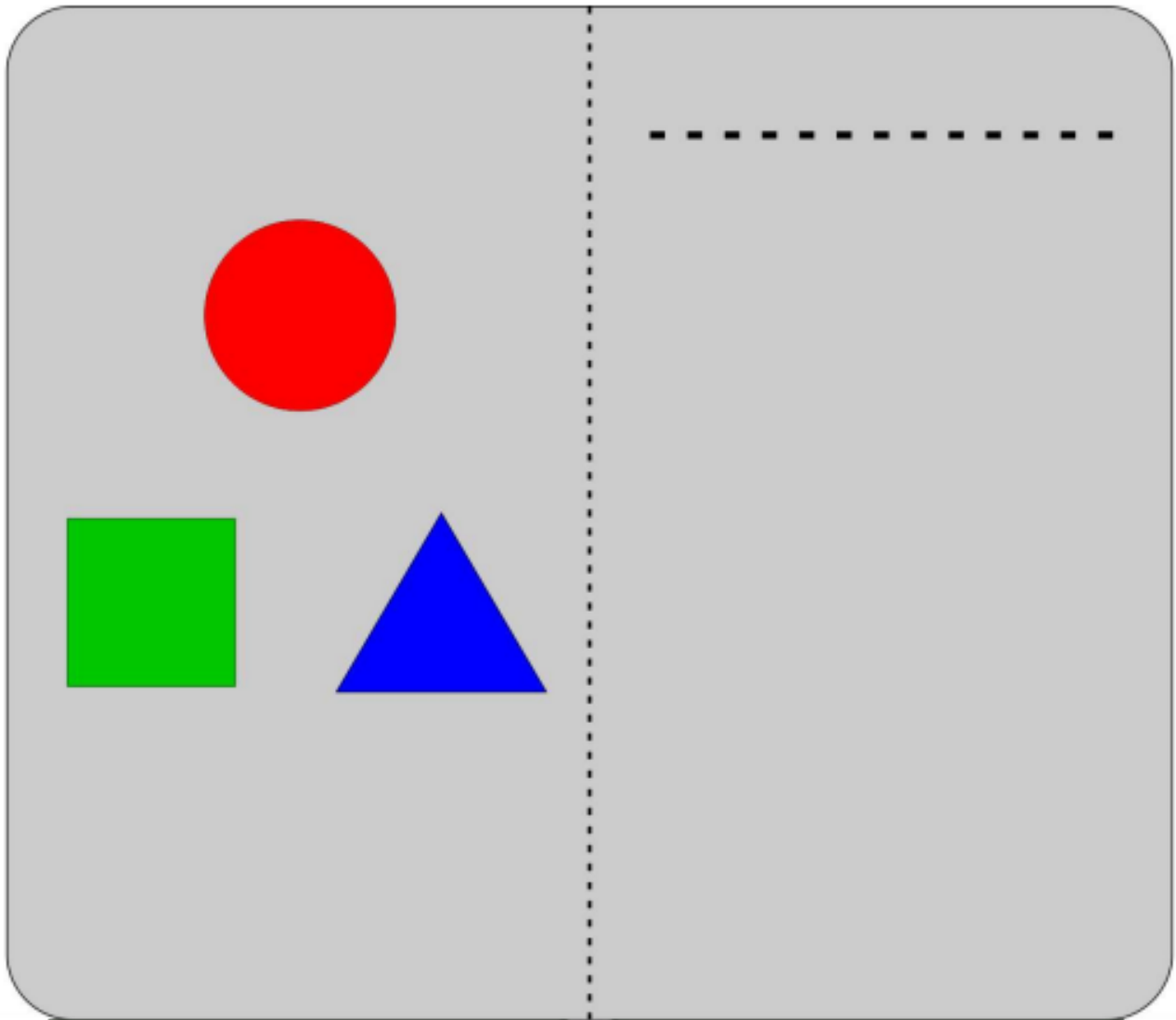
Instruções:

- Entregar uma folha para cada dupla de estudantes;
- Todas as peças devem ser recortadas, Figuras 114 e 115, (um controle e oito cartões de programas);
- Cada crianças deve ficar com quatro cartões de programas e preenchê-los com ações que sejam simples de serem executadas (e.g. pular, bater palmas, girar);
- Após as cartas serem preenchidas, uma delas fica com o controle remoto e insere o cartão na fenda traseira e “instala” o programa;
- O outro membro da dupla, então, por sua vez, deve seguir os comandos que constam no programa (cartão);
- Após alguns instantes, a dupla troca de posição e o outro colega passa a executar os seus programas, inserindo novamente na fenda e instalando os comandos;

• Caso os estudantes não tenham preenchido todos os cartões, encoraje-os a criar novas instruções.

Folha com as imagens a serem recortadas:

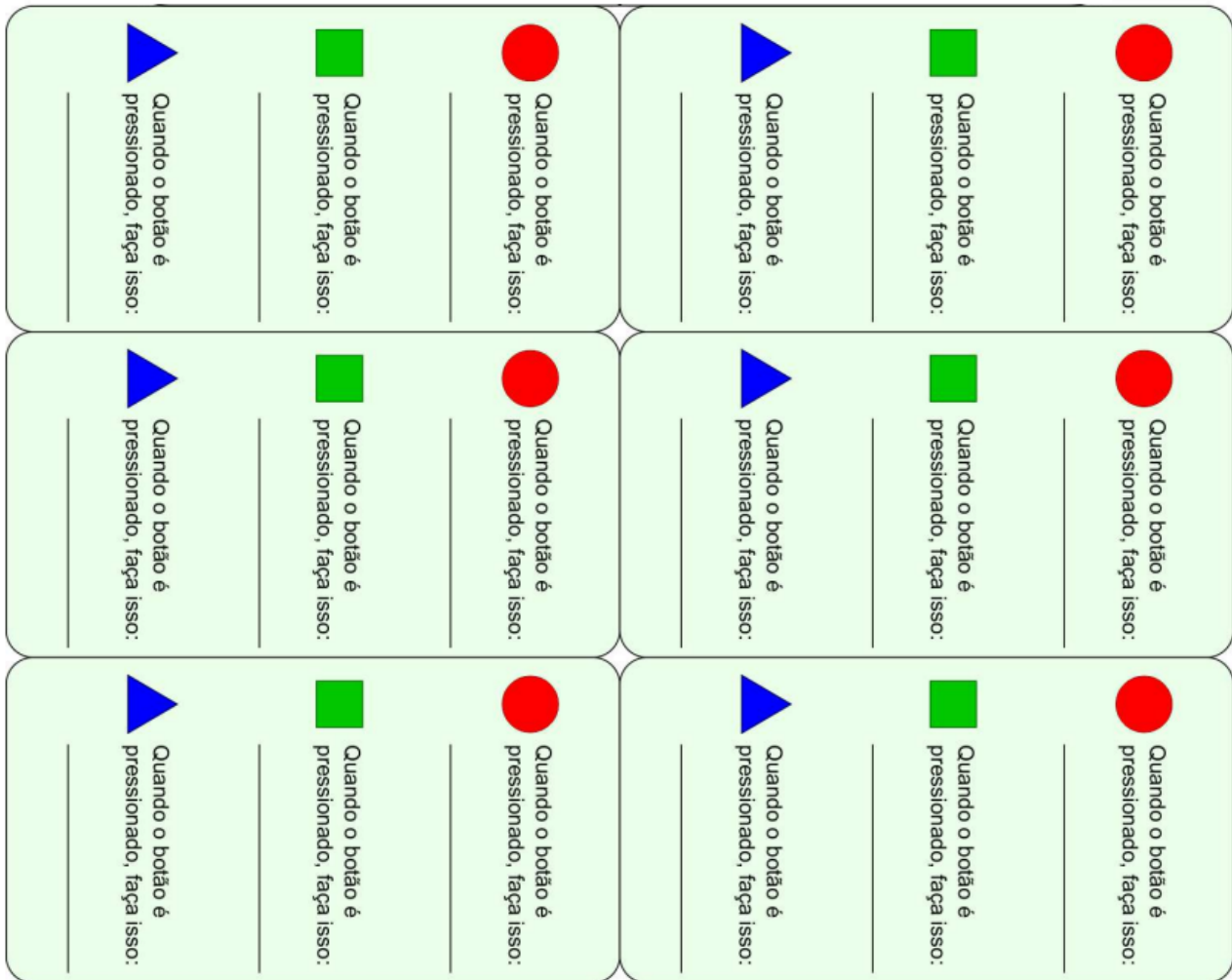
Figura 114 - Cartões controle remoto, recorte 1



Fonte: Site UFRGS<sup>71</sup>

<sup>71</sup> UFRGS - Universidade Federal do Rio Grande do Sul. Adaptado de: <<https://lume.ufrgs.br/bitstream/handle/10183/172208/001054290.pdf?sequence=1&isAllowed=y>>.pag. 125. Acesso em 03/2022.

Figura 115 - Cartões controle remoto, recorte 2



Fonte: Site UFRGS

### 6.2.1.2 Oficina didática - Colorindo com números

O objetivo dessa oficina didática é explicar como os computadores podem armazenar imagens usando somente números. A mesma foi baseada na atividade “Colorindo com Números” de [Bell *et al.* 2005].

Conceitos: Representação de imagens.

Descrição da atividade: O conceito de pixels pode ser explicado usando as imagens, que para simplificar, são pretas e brancas. Os alunos deverão aprender como representar as imagens usando números, representando a quantidade de pixels brancos e pretos, como na Figura 116. A qual, nos mostra como uma imagem pode ser representada por números. A primeira linha consiste em um pixel branco, seguido de três pixels pretos e, por fim, de um pixel branco. Assim, a primeira linha é representada por 1, 3, 1.



O primeiro número sempre se refere ao número de pixels brancos. Se o primeiro pixel for preto, a linha começará com um zero. Por exemplo, a quarta linha começa com preto, mas depois fica branca mais três casas e, por último, fica preta novamente. Note que antes de colocar o número de casas brancas, devemos usar o número 1 para representar esta cor.

Materiais: Folha impressa quadriculada conforme a Figura 117 - Cartão crie uma imagem. Cada estudante deverá ter a sua folha de atividade.

Tempo estimado: De 10 a 30 minutos.

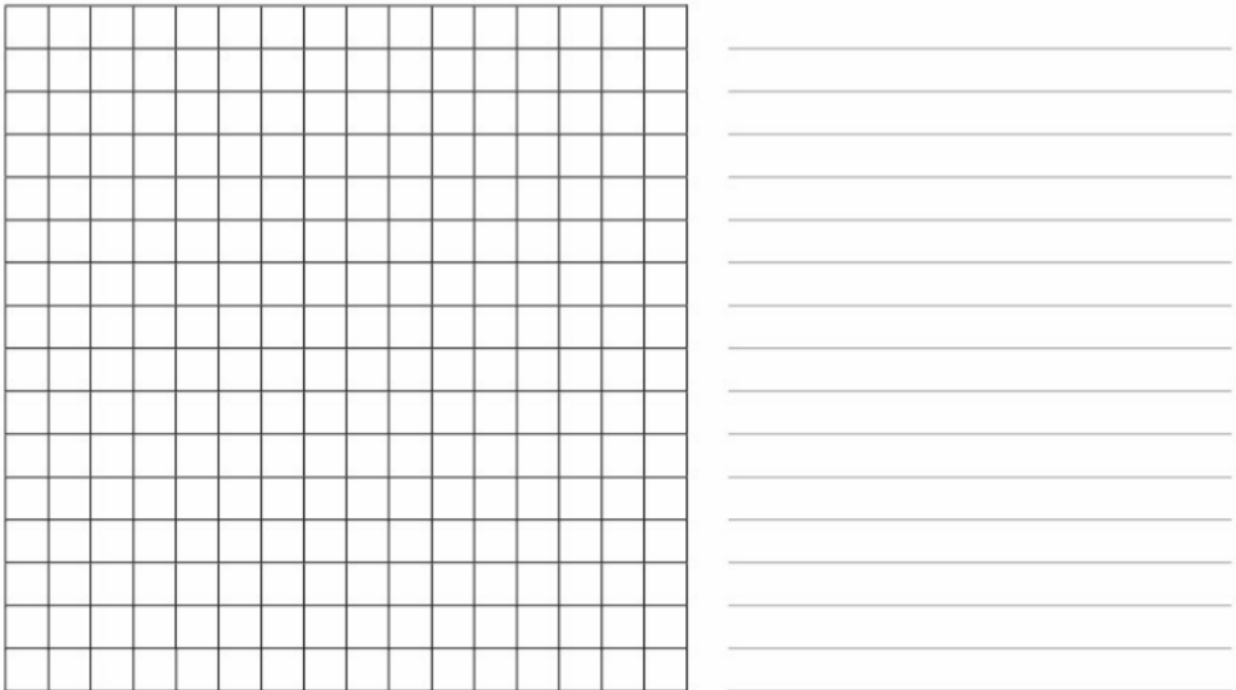
**Figura 116 - Cartão colorindo com números**

	■	■	■		1, 3, 1
				■	4, 1
	■	■	■	■	1, 4
■				■	0, 1, 3, 1
■				■	0, 1, 3, 1
	■	■	■	■	1, 4

Fonte: Adaptada de dinâmicas e atividades desplugadas de UFRJ<sup>72</sup>.

<sup>72</sup> UFRJ - Universidade Federal do Rio de Janeiro, Campus Macaé. Adaptado de: <<https://aprendaprogramar.macaee.ufrj.br/>>. pag. 12. Acesso em 03/2022.

**Figura 117 - Cartão crie uma imagem**



Fonte: Adaptada de dinâmicas e atividades desplugadas (p. 33) de UFRJ.

### 6.2.1.3 Oficina didática - Passeio da tartaruga

O objetivo dessa oficina didática é mostrar a forma com a qual o computador interpreta um conjunto de coordenadas dadas. Para isso, a atividade simula em um quadro o movimento de uma tartaruga.

Conceitos: Comando de repetição, algoritmo e movimento.

Materiais:

- Blocos do scratch, Figura 118, impressos em papel. Para visualizar cada um dos blocos escolha a categoria, acesse:

<https://scratch.mit.edu/projects/722289649/fullscreen/> ou ainda crie suas próprias figuras em <https://printscratch.com/>:

3x “virar à direita”; 4x “ir para frente”; 1x “repetir”;

- Um desenho de uma tartaruga, Figura 119, para um estudante segurar;
- 1 caneta de quadro branco.

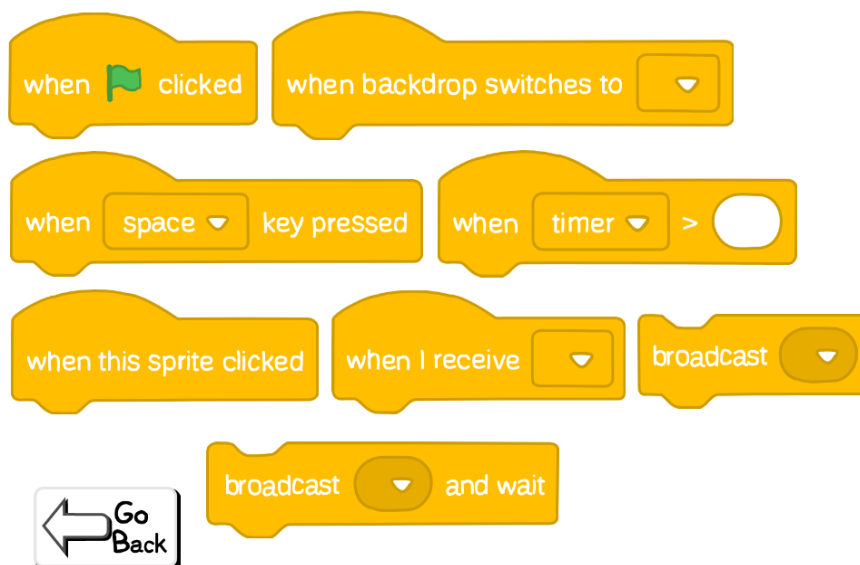
Descrição da atividade:

1. Recorte a imagem, Figura 119 - Passeio da tartaruga, e cole a imagem na parte de trás da caneta de quadro;

2. Escolha com os alunos qual deve ser o caminho que a tartaruga irá desenhar. Os desenhos podem ser figuras geométricas ou caminhos específicos, como uma escada ou um labirinto. Coloque o desenho em um lado do quadro;
3. Escreva os comandos que podem ser ditos e explique que os programas de computadores apenas aceitam os comandos que conhecem;
4. Os alunos dão as ordens (vire para a direita, ande 1 passo...) para que um desenho se forme no quadro. A cada comando que o aluno falar realize o desenho no quadro;
5. Supondo o caso do jogo iniciando com a tartaruga virada para a direita na horizontal, um exemplo do que seria esperado que o aluno respondesse para formar uma escada seria:
6. “Ande 1 passo, vire 90 graus para a esquerda e ande 1 passo. Em seguida, vire para a direita e ande 1 passo. Agora, vire para a esquerda e ande 1 passo...”.
7. A partir disso, o instrutor poderia oferecer uma solução mais prática para realizar o mesmo desenho, utilizando um comando o comando de repetição:
8. “Repita 3 vezes o seguinte: ande, vire para a esquerda, ande e vire para a direita”.
9. Diversos outros desenhos poderiam ser feitos com a mesma ideia.

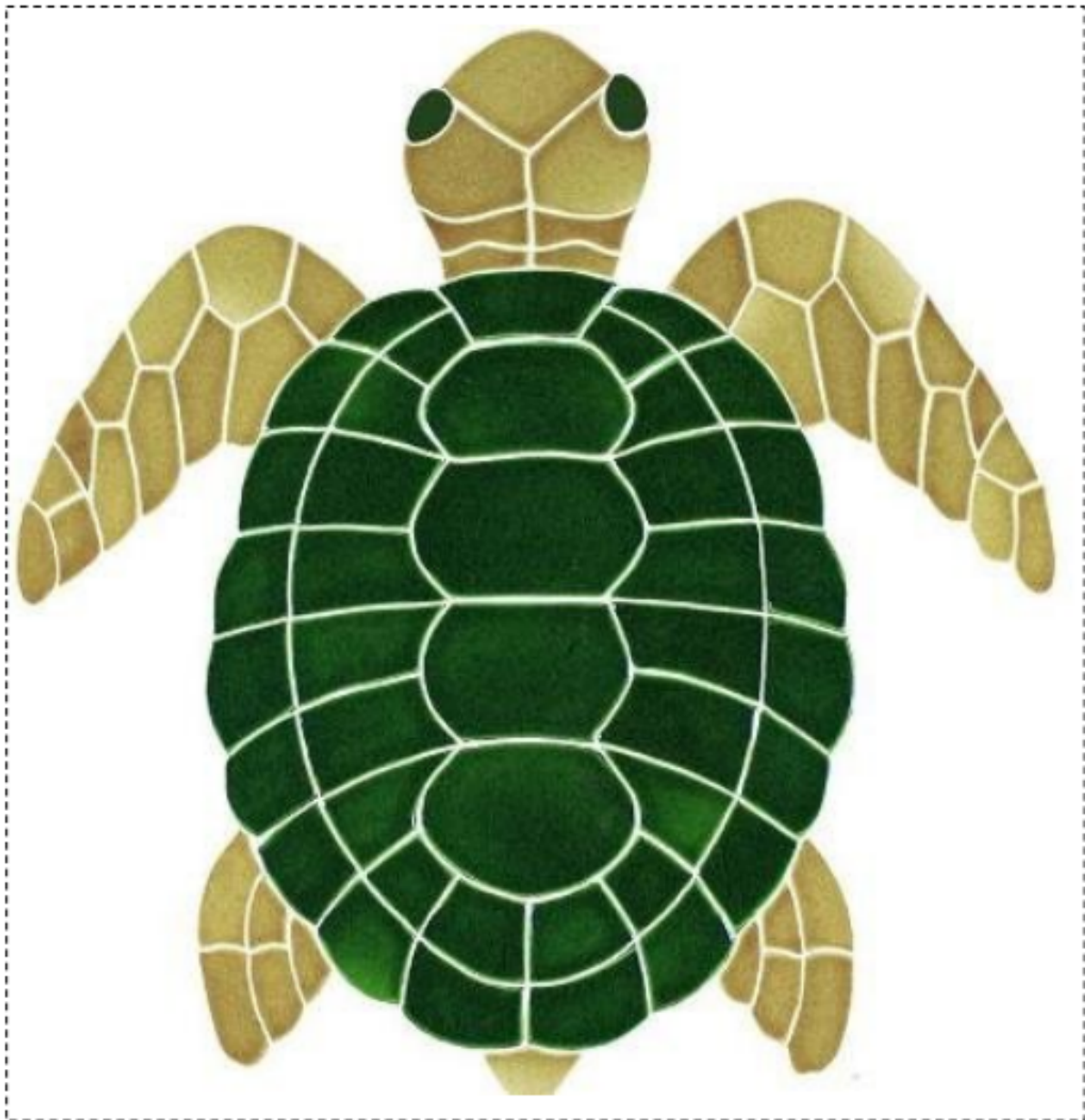
Figura 118 - Blocos Scratch - Passeio da tartaruga

### Events Blocks:



Fonte: Scratch MIT. Disponíveis em: <https://scratch.mit.edu/projects/722289649>

Figura 119 - Passeio da tartaruga



Fonte: Adaptada de dinâmicas e atividades desplugadas (p. 34) UFRJ<sup>73</sup>.

<sup>73</sup> UFRJ - Universidade Federal do Rio de Janeiro, Campus Macaé. Adaptado de: <<https://aprendaprogramar.macaee.ufrj.br/>>, p. 34. Acesso aqui em 03/2022.

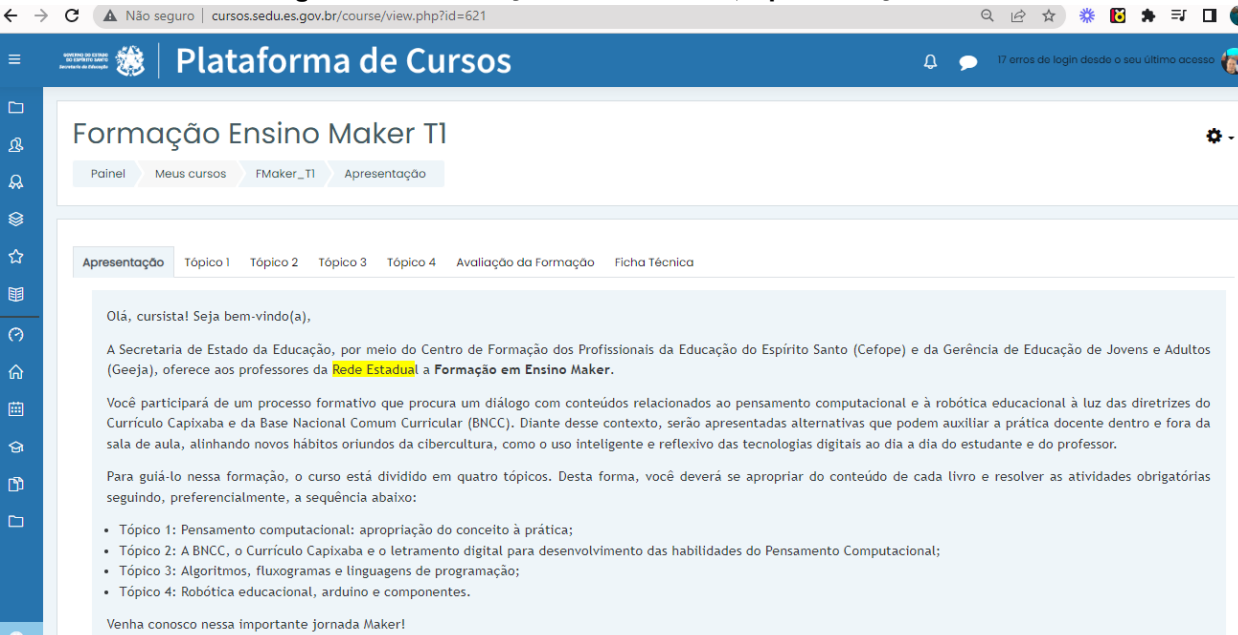
## 7. O Curso de formação de profissionais da educação - Formação Ensino Maker

O curso de formação de profissionais da educação, Formação Ensino Maker, foi resultado do produto das pesquisas realizadas para escrever este trabalho de dissertação, baseada na necessidade do estudante e dos profissionais da educação e também na literatura bibliográfica sobre o assunto que deu embasamento para a criação e produção de materiais e recursos para a construção dos materiais e recursos para o ambiente virtual ( <http://cursos.sedu.es.gov.br/> ).

A partir da parceria juntamente com a equipe SEDU-CEFOPE, Centro de Formação dos Profissionais da Educação do Espírito Santo que foi fundamental nesse processo de construção que se deu por intermédio de reuniões com diálogo individuais e coletivos pela qual foi firmado a proposta da seguinte maneira e distribuição da formação no ambiente do curso:

- Apresentação, Figura 120;

**Figura 120 - Formação Ensino Maker, Apresentação**



The screenshot displays the 'Plataforma de Cursos' interface. The browser address bar shows 'cursos.sedu.es.gov.br/course/view.php?id=621'. The page title is 'Formação Ensino Maker T1'. The breadcrumb trail is 'Painel > Meus cursos > FMaker\_T1 > Apresentação'. The main content area features a navigation menu with 'Apresentação', 'Tópico 1', 'Tópico 2', 'Tópico 3', 'Tópico 4', 'Avaliação da Formação', and 'Ficha Técnica'. The 'Apresentação' section contains a welcome message: 'Olá, cursista! Seja bem-vindo(a),'. It also states: 'A Secretaria de Estado da Educação, por meio do Centro de Formação dos Profissionais da Educação do Espírito Santo (Cefope) e da Gerência de Educação de Jovens e Adultos (Geeja), oferece aos professores da Rede Estadual a Formação em Ensino Maker.' The course description mentions a formative process with topics related to computational thinking and educational robotics. A list of topics is provided: 'Tópico 1: Pensamento computacional: apropriação do conceito à prática;', 'Tópico 2: A BNCC, o Currículo Capixaba e o letramento digital para desenvolvimento das habilidades do Pensamento Computacional;', 'Tópico 3: Algoritmos, fluxogramas e linguagens de programação;', and 'Tópico 4: Robótica educacional, arduino e componentes.' The page concludes with 'Venha conosco nessa importante jornada Maker!'.

Fonte: Autoria própria

- Tópico 1: Pensamento computacional: apropriação do conceito à prática, Figura 121;

**Figura 121 - Formação Ensino Maker, tópico 1**

The screenshot shows a web browser window with the URL `courses.sedu.es.gov.br/course/view.php?id=621&section=1`. The page title is 'Plataforma de Cursos'. The main content area is titled 'PENSAMENTO COMPUTACIONAL APROPRIAÇÃO DO CONCEITO À PRÁTICA'. It includes a greeting to the student, a description of the topic's objective, and a list of activities: 'Leia o Livro 1;' and 'Responda o Questionário do Tópico 1.'. Below this, there is a section for 'MATERIAIS DE ESTUDO' with a link to '[Livro 1] - Pensamento computacional: apropriação do conceito à prática'.

Fonte: Autoria própria

- Tópico 2, Figura 122: A BNCC, o Currículo Capixaba (Currículo do ES) e o letramento digital para desenvolvimento das habilidades do Pensamento Computacional;

**Figura 122 - Formação Ensino Maker, tópico 2**

The screenshot shows a web browser window with the URL `courses.sedu.es.gov.br/course/view.php?id=621&section=2`. The page title is 'Plataforma de Cursos'. The main content area is titled 'A BNCC, O CURRÍCULO CAPIXABA LETRAMENTO DIGITAL'. It includes a greeting to the student, a description of the topic's objective, and a list of activities: 'Leia o Livro 2;' and 'Responda o Questionário do Tópico 2.'. Below this, there is a section for 'MATERIAIS DE ESTUDO' with a link to '[Livro 2] - A BNCC, o Currículo Capixaba e o letramento digital para desenvolvimento das habilidades do PC' and a section for 'ATIVIDADES' with a link to '[Atividade 2] - Questionário'.

Fonte: Autoria própria.

- Tópico 3, Figura 123: Algoritmos, fluxogramas e linguagens de programação;

**Figura 123 - Formação Ensino Maker, tópico 3**

The screenshot shows a web browser window with the URL `courses.sedu.es.gov.br/course/view.php?id=621&section=3`. The page title is 'Plataforma de Cursos'. The navigation menu includes 'Apresentação', 'Tópico 1', 'Tópico 2', 'Tópico 3' (selected), 'Tópico 4', 'Avaliação da Formação', and 'Ficha Técnica'. The main content area features a green header with a flowchart icon and the text 'ALGORITMOS, FLUXOGRAMAS E LINGUAGENS DE PROGRAMAÇÃO'. Below the header, there is a message to the student: 'Prezado(a) cursista, Esse tópico tem por objetivo apresentar os conceitos de fluxograma e algoritmo, segundo a BNCC e a SBC, além de apresentar alguns dos tipos de linguagens de programação existentes mais utilizados e abordar quais deles seriam mais adequados para o desenvolvimento de um determinado projeto.' This is followed by 'ATIVIDADES DO TÓPICO 3' with a list: 'Leia o Livro 3;' and 'Responda o Questionário do Tópico 3.' Below this is 'Bons estudos!'. At the bottom, there are sections for 'MATERIAIS DE ESTUDO' (listing '[Livro 3] - Algoritmos, fluxogramas e linguagens de programação') and 'ATIVIDADES' (listing '[Atividade 3] - Questionário').

Fonte: Autoria própria

- Tópico 4, Figura 124: Robótica educacional, arduino e componentes;

**Figura 124 - Formação Ensino Maker, tópico 4**

The screenshot shows a web browser window with the URL `courses.sedu.es.gov.br/course/view.php?id=621&section=4`. The page title is 'Plataforma de Cursos'. The navigation menu includes 'Apresentação', 'Tópico 1', 'Tópico 2', 'Tópico 3', 'Tópico 4' (selected), 'Avaliação da Formação', and 'Ficha Técnica'. The main content area features a green header with a microchip icon and the text 'ROBÓTICA EDUCACIONAL ARDUÍNO E COMPONENTES'. Below the header, there is a message to the student: 'Prezado(a) cursista, Esse tópico tem por objetivo apresentar alguns conceitos e definições sobre a temática da Robótica Educacional, propondo formas de utilização em sala de aula da placa e emulador de Arduino e seus componentes básicos. Na oportunidade, será apresentado o Tinkercad, uma ferramenta digital gratuita que permite reproduzir tais experimentos e, a partir disso, simular o comportamento dos circuitos. Por fim, o tópico trará proposta de sites, aplicativos, oficinas e aplicações/atividades da computação plugada e desplugada para serem utilizados dentro e fora da sala de aula.' This is followed by 'ATIVIDADES DO TÓPICO 4' with a list: 'Leia o Livro 4;' and 'Responda o Questionário do Tópico 4.' Below this is 'Bons estudos!'. At the bottom, there are sections for 'MATERIAIS DE ESTUDO' (listing '[Livro 4] - Robótica educacional, arduino e componentes') and 'ATIVIDADES' (listing '[Atividade 4] - Questionário').

Fonte: Autoria própria.

- Avaliação da Formação, Figura 125.

Figura 125 - Formação Ensino Maker, avaliação da formação



Fonte: A autoria própria



## 8. Conclusão

Ao longo da trajetória das pesquisas e da escrita deste trabalho, assim como o desenvolvimento das oficinas didáticas, muito aprendizado aconteceu neste produto de pesquisas através das consultas bibliográficas que teve como objetivo a preparação da proposta de formação de professores para o desenvolvimento das habilidades e competências do PC, alinhados à BNCC e ao Currículo do ES. Assim como, no estudo do Currículo do ES e da BNCC, considerações relevantes de autores especialistas, têm motivado nas buscas e alternativas que podem auxiliar os professores em sua prática dentro e fora de sala de aula, com a proposta de mediar ou nortear os profissionais por meio de apropriação dos conceitos apresentados e implementar em seus planos de aula e/ou sequências didáticas, assim também para implementação de projetos, experimentos e apresentação de ferramentas para seu desenvolvimento pessoal e profissional. Dentro da proposta, o produto, curso de formação de professores, Formação Ensino Maker, em parceria com a equipe da SEDU - Secretaria de Estado da Educação do Espírito Santo, teve cunho fundamental nesse processo de construção e desenvolvimento dos materiais, assim como, a aceitação com entusiasmo pelos professores que se inscreveram. Nesse processo, tivemos 1356 inscrições na Formação Ensino Maker na qual estavam previstas apenas 400 vagas. Estas inscrições para a formação aconteceram de forma voluntária com resultados positivos na aceitação dos recursos propostos e pode mostrar o quanto essa temática é esperada nas formações de professores.

Nesse sentido, para estudos posteriores, seria válida enriquecer essa abordagem e apropriação do pensamento computacional e da robótica educacional com planejamento de um currículo mais incisivo sobre a competência 5 da BNCC para uma maior apropriação das novas tendências das tecnologias educacionais e dessa forma facilitar por meio de planejamentos e estratégias a compreensão, a utilização e a criação de tecnologias digitais de informação e comunicação possibilitando por meios destas a produção de conhecimento, a resolução de problemas simples e complexos no mundo do trabalho e assim, abrindo caminho para o protagonismo e autoria de modo a valorizar a autonomia a consciência crítica e responsável fortalecendo assim a relação com o mundo e suas necessidades.

Ainda pensando nas possibilidades, mais recente surgiu o ChatGPT que é uma tecnologia baseada em inteligência artificial capaz de imitar a linguagem humana, e pode gerar respostas a partir de informações e dados pré-existentes fornecidos pelos usuários. É o futuro da tecnologia, considerando os avanços nas linguagens de programação e a necessária atualização dos currículos escolares para uma escola formadora mais moderna e atualizada.

## 9. Referências

ALENCAR, L. *et al.* (2020). **Criação de um jogo para desenvolver o pensamento computacional percorrendo caminhos eulerianos.** In WEI, pages 111–115. SBC. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/11140>>. Acesso em: 03/2022.

ALGORITMO. *In:* DICIO, Dicionário online, **Michaelis**. Disponível em: <<https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/algoritmo/>>. Acesso em 09/2022.

ALMEIDA, M. A. **Possibilidades da robótica educacional para a educação matemática.** Dia a dia Educação. Curitiba – PR. 2007. Disponível em: <<http://www.diaadiaeducacao.pr.gov.br/portals/pde/arquivos/363-4.pdf>>. Acesso em 05/2022.

ANDRADE, Daiane; CARVALHO, Tainã; SILVEIRA, Jayne; CAVALHEIRO, Simone; FOSS, Luciana; FLEISCHMANN, Ana Marilza; AGUIAR, Marilton; REISER, Renata. Proposta de Atividades para o Desenvolvimento do Pensamento Computacional no Ensino Fundamental. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 19. , 2013, Campinas. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2013 . p. 169-178. Disponível em: < <https://sol.sbc.org.br/index.php/wie/article/view/16658>>. Acesso em 06/2023.

ARDUINO. **O que é o Arduino?**. 2019. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em 05/2022.

BACICH, Lilian; MORAN, José (Orgs.). **Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática.** Porto Alegre: Penso, 2018. Disponível em: <<https://curitiba.ifpr.edu.br/wp-content/uploads/2020/08/Metodologias-Ativas-para-uma-Educacao-Inovadora-Bacich-e-Moran.pdf>>. Acesso em 05/2022.

BACICH, Lilian; TANZINE NETO, Adolfo ;TREVISANI , Fernando . **Ensino Híbrido : Personalização e Tecnologia na Educação**. Porto Alegre: Penso, 2015. Disponível em: <<http://www.senar-rio.com.br/wp-content/uploads/2021/03/Livro-Ensino-H%C3%ADbrido.pdf>>. Acesso em 05/2022.

BARCELOS, T. S. e SILVEIRA, I. F. (2012). “Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica”. In: **Anais do Congresso da Sociedade Brasileira de Computação**, Curitiba, SBC, p.1-10. Disponível em: <[https://www.academia.edu/43649113/Anais\\_do\\_Congresso\\_Sertanejo\\_de\\_Computa%C3%A7%C3%A3o](https://www.academia.edu/43649113/Anais_do_Congresso_Sertanejo_de_Computa%C3%A7%C3%A3o)>. Acesso em 03/2022.

BAUDSON, Adolfo José Gonçalves Stavaux. ARAÚJO, Francisco César Rodrigues de. **Rede e-tec Brasil**. Algoritmos e Programação. 2013. Disponível em: <<https://www.ifmg.edu.br/ceadop3/apostilas/algoritmos-e-programacao>>. Acesso em 02/2022.

BEASANT, Pam. **Introdução à eletrônica**. 1985. Copyright by Usborne Publishing Ltd. Tradução de Editora Lutécia LTDA. Disponível em: <<https://8book.com.br/user/pages/pdf/curso-de-eletronica-ilustrado-para-iniciantes/Curso-de-Eletronica-Ilustrado.pdf>>. Acesso em 05/2022.

BEER, Raquel. Programação para menores. **Veja**, São Paulo, ed. 2329, p. 86-89, 10 jul. 2013. Disponível em: <<https://liag.ft.unicamp.br/wp-content/uploads/2015/12/Scratch-veja-2013.pdf>>. Acesso em 01/2022.

BELL, T., WITTEN, I, H., FELLOWS M. (2011). **Computer Science Unplugged: Ensinando Ciência da Computação sem o uso do computador**. Tradução coordenada por Luciano Porto Barreto. Disponível em:<<https://classic.csunplugged.org/documents/books/portuguese/CSUnpluggedTeachers-portuguese-brazil-feb-2011.pdf>>. Acesso em 03/2022.

BERRI, R.A.; GRASSI Jr.,V.; OSORIO, F. S. "Simulação de Robôs Móveis e Articulado: Aplicações e Prática". (Capítulo 6). 34a. JAI - Jornada de Atualização de Informática, **XXXV Congresso da SBC**, Recife - PE, pp.274-322. Julho 2015. Disponível em: <<https://www.sites.google.com/site/vrepjai/>>. Acesso em 01/2022.

BLIKSTEIN, P.(2008) “**O pensamento computacional e a reinvenção do computador na educação**”. Disponível em: <[http://www.blikstein.com/paulo/documents/online/ol\\_pensamento\\_computacional.html](http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html)>. Acesso em 03/2022.

BOCCONI, S., CHIOCCARIELLO, A., DETTORI, G., FERRARI, A., ENGELHARDT, K., KAMPYLIS, P., & PUNIE, Y. (2016). Desenvolvendo o pensamento computacional no ensino obrigatório. **Comissão Europeia, JRC Science for Policy Report** , 68. Disponível em: <<https://publications.jrc.ec.europa.eu/repository/handle/JRC104188?mode=full>>. Também Disponível em pdf: <[https://komenskypost.nl/wp-content/uploads/2017/01/jrc104188\\_computhinkreport.pdf](https://komenskypost.nl/wp-content/uploads/2017/01/jrc104188_computhinkreport.pdf)>. Acesso em 03/2022.

BRASIL (2017). BNCC - Base nacional comum curricular. **Ministério da Educação**. 2017. Disponível em: <[http://basenacionalcomum.mec.gov.br/images/BNCC\\_EI\\_EF\\_110518\\_versaofinal\\_site.pdf](http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf)>. Acesso em 03/2022.

BRACKMANN, C. P. (2017). **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brasil. Disponível em: <<https://lume.ufrgs.br/handle/10183/172208>>. Acesso em 03/2022.

BUNDY, A. 2007 O pensamento computacional é difundido. **J. Científico**. Pratique. Computar. 1 , 67-69. Disponível em: <Google Scholar>. Acesso em 03/2022.

CARVALHO, A. L. **Apostila Arduino: exemplos de aplicação**. São Paulo, 2016. Apostila resumida com aplicações para iniciar um trabalho na Plataforma Arduino. Disponível em: <<https://www.passeidireto.com/arquivo/61171296/carvalho-alexandre-apostila-arduino-exemplos-aplicacao>>. Acesso em 05/2022.

COLL, C. (2018). Apresentação e prólogo do livro "Pensamento computacional. Análise de uma competência-chave". **Revista de Educação a Distância** , 19 . Disponível em: <<https://revistas.um.es/red/article/view/395281>>. Acesso em 03/2022.

CORDEIRO, Filipe. App Inventor: Guia de Criação de Apps. **Blog AndroidPro**. [S.l], [2017]. Disponível em: <<https://www.androidpro.com.br/blog/desenvolvimento-android/appinventor/>>. Acesso em 05/2022.

ESPÍRITO SANTO (Estado). Secretaria de Estado da Educação. **Currículo do ES**. 2020. Disponível em: <<https://curriculo.sedu.es.gov.br/curriculo/documentos/>>. Acesso em 11/2021.

CSTA, ISTE, and NSF (2010). **Computational thinking leadership toolkit**. <<http://www.csta.acm.org/Curriculum/sub/CompThinking.html>>. Acesso em 03/2022.

FERNANDES, Rebeca Chiacchio Azevedo. Inovações Pedagógicas no Ensino de Ciências dos Anos Iniciais: um estudo a partir de pesquisas acadêmicas brasileiras (1972-2012). 2015. 397p. **Tese (Doutorado em Educação)** – Faculdade de Educação, Universidade Estadual de Campinas, Campinas, 2015. Disponível em: <<https://core.ac.uk/download/pdf/296883196.pdf>>. Acesso em 09/2022.

FERREIRA, Williane Costa; DE OLIVEIRA, Carloney Alves. O Scratch nas aulas de matemática: caminhos possíveis no ensino das áreas de figuras planas. **Cadernos Cenpec| Nova série**, v. 8, n. 1, 2018. Disponível em: <<https://cadernos.cenpec.org.br/cadernos/index.php/cadernos/article/view/390>>. Acesso em 02/2022.

FLUXOGRAMA. *In*: DICIO, Dicionário online, **Michaelis**. Disponível em: <<https://michaelis.uol.com.br/palavra/OVW2/fluxograma/>>. Acesso em 09/2022.

FRANÇA, R. S. e AMARAL, H. J. C. (2013). “Proposta metodológica de ensino e avaliação para o desenvolvimento do pensamento computacional com o uso do Scratch”. *In*: **Anais do Congresso Brasileiro de Informática na Educação**, Campinas, SBC, p. 179–188.

FÓRUM ECONÔMICO MUNDIAL (WEF). **Relatório O Futuro dos Empregos e as habilidades do futuro**. Disponível em: <<https://www.weforum.org/reports/the-future-of-jobs-report-2020>>. Acesso em 03/2022.

FUENTES, Rodrigo Cardozo. Eletrônica / Rodrigo Cardozo Fuentes. – 3. ed. – Santa Maria : **Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Curso Técnico em Automação Industrial**, 2009. Disponível em: <[http://redeetec.mec.gov.br/images/stories/pdf/eixo\\_ctrl\\_proc\\_indust/tec\\_autom\\_ind/eletronica/161012\\_eletronica.pdf](http://redeetec.mec.gov.br/images/stories/pdf/eixo_ctrl_proc_indust/tec_autom_ind/eletronica/161012_eletronica.pdf)>. Acesso em 05/2022.

GAROFALO, D. Como levar o STEAM para a sala de aula. **Nova Escola**, 25, jun. de 2019. Disponível em: <<https://novaescola.org.br/conteudo/18021/como-levar-o-steam-para-a-sala-de-aula>>. Acesso em 11/2022.

KLEIMAN, A. B & Assis, J. A. (Orgs). Significados e Ressignificações do Letramento: desdobramentos de uma perspectiva sociocultural sobre a escrita. 1ª edição. Campinas: **Mercado de Letras**, 2016.

KLEIMAN, Angela B. (org), Os significados do letramento: uma nova perspectiva sobre a prática social da escrita. Campinas: **Mercado de Letras**, 1995.

LAUER, A.J.; Cordenonsi, A.Z. **Oficinas de Arduino: Manual para Professores de Lógica de Programação**. Santa Maria: 2020. Relatórios Técnicos do Programa de Pós-Graduação em Tecnologias Educacionais em Rede, v. 2., n.1. Disponível em: <[https://repositorio.ufsm.br/bitstream/handle/1/22335/DIS\\_PPGTER\\_2020\\_LAUER\\_ANDERSON.pdf?sequence=1](https://repositorio.ufsm.br/bitstream/handle/1/22335/DIS_PPGTER_2020_LAUER_ANDERSON.pdf?sequence=1)>. Acesso em 11/2022.

LIMA, Elielson Magalhães. *Et al.* Robotics and Mathematics Teaching: Building a simple traffic light in the classroom. **Revista Científica Multidisciplinar Núcleo do Conhecimento**. Year 05, Ed. 12, Vol. 18, pp. 67-77. Dezembro 2020. ISSN: 2448-0959. Disponível em: <<https://www.nucleodoconhecimento.com.br/educacao/construindo-um-semaforo>>. Acesso em 11/2022.

LIMA, M. R. de. (2018). Concepções docentes acerca do ensino de programação de computadores no ensino superior. **Teoria e Prática da Educação**. Disponível em: <<https://periodicos.uem.br/ojs/index.php/TeorPratEduc/article/view/45599>>. Acesso em 09/2022.

LOPES, Alice Casimiro; MACEDO, Elizabeth. **Teorias de currículo**. São Paulo: Cortez, 2011. Disponível em: <[https://edisciplinas.usp.br/pluginfile.php/3225299/mod\\_resource/content/1/Lopes%20e%20Macedo%20capitulo3\\_Conhecimento.pdf](https://edisciplinas.usp.br/pluginfile.php/3225299/mod_resource/content/1/Lopes%20e%20Macedo%20capitulo3_Conhecimento.pdf)>. Acesso em 01/2022.

MACEDO, Elizabeth. **Currículo e conhecimento: Aproximações entre educação e ensino**. 2012. Disponível em: <<https://www.scielo.br/j/cp/a/GfnkdSkSTRY6TgSPLmYYz8K/?format=pdf&lang=pt>>. Acesso em 03/2022.

MACEDO, Lino de. **Ensaio pedagógico: Como construir uma escola para todos?** Porto Alegre: Artmed, 2005. Disponível em: <<https://www.nescon.medicina.ufmg.br/biblioteca/imagem/2505.pdf>>. Acesso em: 03/2022.

MATARIĆ, Maja J. **The Robotics Primer**. 2007. Disponível em: <<https://pages.ucsd.edu/~ehutchins/cogs8/mataric-primer.pdf>>. Acesso em 05/2022.



MARTINELLI, Rodolfo. Zaina, L. A. M. e SAKATA, T. C. “MultiTACT: Uma abordagem para a construção de atividades de ensino multidisciplinares para estimular o Pensamento Computacional no Ensino Fundamental I”. **Anais** “VII Congresso Brasileiro de informática na Educação (CBIE 2019)”. Disponível em: <<http://gg.gg/MultiTACT-atividades-PC>>. Acesso em 05/2022.

MONK, Simon. **30 projetos com Arduino** [recurso eletrônico]/Simon Monk; tradução: Anatólio Laschuk. 2. ed. - Dados eletrônicos. - Porto Alegre: Bookman, 2014. Disponível em: <<https://jesuegraciliano.files.wordpress.com/2018/06/30-projetos-com-arduino-simon-monk.pdf>>. Acesso em 12/2022.

NÓVOA, A. A formação em foco. **Revista Nova Escola**. São Paulo: Ed. Abril, n. 142, maio 2001. Disponível em: <[https://ledum.ufc.br/arquivos/didatica/3/Professor\\_Pesquisador\\_Reflexivo.pdf](https://ledum.ufc.br/arquivos/didatica/3/Professor_Pesquisador_Reflexivo.pdf)>. Acesso em 01/2022.

NÓVOA, António. Universidade de Lisboa, Portugal. **Os Professores e a sua Formação num Tempo de Metamorfose da Escola**. Educação & Realidade, vol. 44, núm. 3, e84910, 2019. UFRGS - Faculdade de Educação. Disponível em: <<https://www.redalyc.org/journal/3172/317265189003/html/>>. Acesso em 03/2022.

NIED. **Robótica Pedagógica**. Núcleo de Informática Aplicada à Educação, [20--?]. Disponível em: <<https://www.nied.unicamp.br/projeto/robotica-pedagogica/>>. Acesso em 05/2022.

NUNES, D. J. (2011). “**Ciência da computação na educação básica**”. Disponível em: <<http://gestaouniversitaria.com.br/artigos/ciencia-da-computacao-na-educacao-basica--3>>. Acesso em 02/2022.

OLIVEIRA, Cláudio Luís Vieira Oliveira. **Aprenda Arduino – Uma abordagem prática** / [texto de Cláudio Luís Vieira Oliveira, Humberto Augusto Piovesana ZANETTI, Cristina Becker Matos NABARRO e Júlio Alberto Vansan GONÇALVES]. – Duque de Caixas: Katzen Editora, 2018. Disponível em: <<http://www.fatecjd.edu.br/fatecino/material/ebook-aprenda-arduino.pdf>>. Acesso em 05/2022.

OLIVEIRA, Sérgio de; PEREIRA, Marconi de Arruda; TEIXEIRA, Fernando A. MIT App Inventor como Ambiente de Ensino de Algoritmos e Programação. In: Workshop (WEI), 2021. **Anais** [...]. Porto Alegre: SBC, 2021. p. 61-70. Disponível em: <<https://doi.org/10.5753/wei.2021.15897>>. Acesso em 07/2023.

PAPERT, Seymour. **A máquina das crianças: repensando a escola na era da informática**. Porto Alegre: Artmed, 2008.

PEREZ, Anderson Luiz Fernandes e DARÓS, Renan Rocha. **Oficina de Robótica**. 2013. Disponível em: <<https://oficinaderobotica.ufsc.br/arduino/>>. Acesso em 01/2022.

PERRENOUD, P. **Dez novas competências para ensinar**. Porto Alegre: Artmed, 2000.

PERRENOUD, Philippe. 2000. **Construindo competências**. Entrevista com Philippe Perrenoud, Universidade de Genebra, por Paola Gentile e Roberta Bencini. Disponível em: <[http://www.unige.ch/fapse/SSE/teachers/perrenoud/php\\_main/php\\_2000/2000\\_31.html](http://www.unige.ch/fapse/SSE/teachers/perrenoud/php_main/php_2000/2000_31.html)>. Acesso em 03/2022.

PETERS, Marcio. Trigonometria: aprendizagem facilitada com o auxílio do software Geogebra e materiais manipuláveis; 2015; **Monografia; Aperfeiçoamento/Especialização em Novas Tecnologias no Ensino da Matemática**. Disponível em: <<http://petersmatematicaufes.blogspot.com/2016/05/trigonometria-aprendizagem-facilitada.html>>. Acesso em 10/2022.

PÓLYA, G.; **How to solve it**. Copyright 1945. Disponível em: <<http://www.im.ufrj.br/~monica/funcoes/Polya.pdf>>. Acesso em 03/2022.

PRENSKY, M.: Digital Natives Digital Immigrants. In: Prensky, Marc. **On the Horizon**. **NCB University Press**, Vol. 9 No. 5, October (2001a). Disponível em <<https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf>>. Acesso em 03/2022.

RAABE, A. L. A., BRACKMANN, C. P., and CAMPOS, F. R. (2018). Currículo de tecnologia e computação. **CIEB**, 2018. Disponível em: <[https://curriculo.cieb.net.br/assets/docs/Curriculo\\_de\\_Referencia\\_em\\_Tecnologia\\_e\\_Computacao.pdf](https://curriculo.cieb.net.br/assets/docs/Curriculo_de_Referencia_em_Tecnologia_e_Computacao.pdf)>. Acesso em 03/ 2022.

ROBÓTICA. In: DICIO, Dicionário online, **Michaelis**. Disponível em: <<https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/robotica/>>. Acesso em 05/2022.

ROCHA, J. J. S.; Götzke A. P. D.; SILVA, I. N.; Biléssimo, S. M. S.; SILVA, J. B. Novas Tecnologias da Informação e Comunicação: uma Análise a partir do Uso de Dispositivos Móveis em uma Escola Multisseriada Indígena. In: **III Simpósio Ibero-Americano de Tecnologias Educacionais**. 2019, Santa Catarina. Anais do SITED 2019. Santa Catarina: UFSC, 2019. p. 265- 273. Disponível em: <<https://publicacoes.rexlab.ufsc.br/index.php/sited/article/view/225>>. Acesso em 01/2022.

SANCHES, William Ferreira. **A Utilização do emulador de arduino tinkercad para o ensino de lógica de programação e eletrônica**. Orientador: Dr. João Ferreira de Santanna Filho. 54 f. 2020. Trabalho de Conclusão de Curso (Licenciatura em Computação) - Universidade Federal Rural da Amazônia, Belém, 2020. Disponível em: <<http://bdta.ufra.edu.br/jspui/handle/123456789/1702>>. Acesso em 06/2022.

SAMPAIO, Silvio Costa. **Lógica para Programação**. Palmas. Editora Educon, 2008.

SBC (2018). Diretrizes para o ensino de computação na educação básica. **Sociedade Brasileira de Computação**. 2018. Disponível em: <<http://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/1177-diretrizes-para-ensino-de-computacao-na-educacao-basica>>. Acesso em 04/2022.

SBC. Diretrizes para ensino de Computação na Educação Básica Ensino de Computação na Educação Básica. **Documento da Sociedade Brasileira de Computação**. 2018a. Disponível em: <<https://www.sbc.org.br/noticias/10-slideshow-noticias/2083-seminario-internacional-debate-computacao-na-educacao-basica>>. Acesso em 12/2020.

SBC. **Nota Técnica da Sociedade Brasileira de Computação sobre a BNCC-EF e a BNCC-EM**. 2018b. Disponível em: <<http://www.sbc.org.br/institucional-3/cartas-abertas/summary/93-cartas-abertas/1197-nota-tecnica-sobre-a-bncc-ensino-medio-e-fundamental>>. Acesso em 12/2022.

SILVA, Alzira Ferreira da. **RoboEduc: uma metodologia de aprendizado com robótica educacional**. 2009. 127 f. Tese (Doutorado em Automação e Sistemas; Engenharia de Computação; Telecomunicações) - Universidade Federal do Rio Grande do Norte, Natal, 2009. Disponível em: <<https://repositorio.ufrn.br/jspui/bitstream/123456789/15128/1/AlziraFS.pdf>>. Acesso em 10/2022.

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. **Referenciais de Formação em Computação: Educação Básica**. Disponível em <<https://www.sbc.org.br/images/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>>. Acesso em 02/2022.

TORRES, A. L. L. et al. **Ferramenta de auxílio no ensino de Organização e Arquitetura de Computadores: extensão Ptolemy para fins educacionais**. n. December 2012. Disponível em: <[http://www2.sbc.org.br/ceacpad/ijcae/v1\\_n1\\_dec\\_2012/IJCAE\\_v1\\_n1\\_dez\\_2012\\_paper\\_3\\_vf.pdf](http://www2.sbc.org.br/ceacpad/ijcae/v1_n1_dec_2012/IJCAE_v1_n1_dez_2012_paper_3_vf.pdf)>. Acesso em 05/2022.

TUCKER, A. (2003). **A Model Curriculum for K–12 Computer Science**. Final Report of the ACM K–12 Task Force Curriculum Committee. Metodologia do Ensino Superior) - Faculdade Visconde de Cairu. Disponível em: <<https://ftp.unpad.ac.id/orari/library/library-ref-eng/ref-eng-3/application/education/curriculum/k12final1022.pdf>>. Acesso em 03/2022.

VALENTE, José Armando (2016). Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. **Revista PUCSP**. Disponível em: <<https://revistas.pucsp.br/index.php/curriculum/article/view/29051/20655>>. Acesso em 03/2022.

VALENTE, J. A. **O salto para o futuro. Cadernos da TV -Escola**. Brasília: Sede MEC, 2005.

\_\_\_\_\_. Diferentes usos do computador na educação. In: \_\_\_\_\_. **Computadores e conhecimento: repensando a educação**. Campinas: NIED-Unicamp, 1993a.

\_\_\_\_\_. Por que o computador na educação. In: VALENTE, J. A. (Org.). **Computadores e conhecimento: repensando a educação**. Campinas: Gráfica da Unicamp, 1993b.

WIKIPÉDIA. **O que é o Arduino**. Disponível em: <<https://pt.wikipedia.org/wiki/Arduino>>. Acesso em 12/2022.

WING, J. (2016). Pensamento computacional - Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. In: **Revista Brasileira de Ensino e Tecnologia**. Disponível em: <<https://periodicos.utfpr.edu.br/rbect/article/view/4711/pdf>>. Acesso em 03/2022.

WING, JM (2006). **Pensamento computacional**. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) , Pittsburgh, PA. ACM 49 (3). Disponível em: <<https://doi.org/10.1109/vlhcc.2011.6070404>>. Acesso em 03/2022.

WING, JM (julho de 2008). **Pensamento computacional e pensamento sobre computação**. A Royal Society Publishing. Disponível em: <<https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2008.0118>>. Acesso em 03/2022.