

4	Botões táteis (Push-button)
1	Buzzer ativo
1	Protoboard
	Cabinhos e fios para ligações

Teste e uso

Ao ligar o circuito, os LEDs piscarão uma vez, o que indicará que o jogo estará no modo fácil de tempo fixo, no entanto, se mantiver o botão de MODO pressionado durante a energização do circuito, o jogo entrará no modo difícil com progressão de tempo, onde o tempo vai diminuindo após cada jogada. Terminada essa inicialização, um dos LEDs piscará juntamente com a emissão de um sinal sonoro e o jogador deverá dar um toque no botão correspondente ao LED que piscou. Se apertar o botão correto, o mesmo LED irá piscar novamente com a emissão de som e logo em seguida outro LED fará o mesmo e o jogador deverá ir pressionando o botão exato de cada LED. Caso o jogador erre e aperte um botão não correspondente à sequência criada, o jogo fará todos os LEDs piscarem com um som de erro.

Código de programação

```
JogoGenius.ino
1  /*****
2  * Projeto: Jogo Genius
3  * Desenvolvedor: Vanderlei Alves
4  * Data: 28/05/2025
5  *****/
6
7  #define BUZZER 10
8  #define MODO 15
9  #define MAX_SEQ 100
10
11 //Verde, Vermelho, Azul, Amarelo
12 const int botoes[] = {2, 3, 4, 5};
13
14 //Verde, Vermelho, Azul, Amarelo
15 const int leds[] = {6, 7, 8, 9};
16
17 int sequencia[MAX_SEQ];
18 int nivel = 0;
19
20 //1 = fixo, 2 = progressivo
21 int modo = 1;
22
23 //Dó, Mi, Sol, Dó agudo
24 const int tons[] = {261, 329, 392, 523};
25 int tempoBase = 500;
26
27 void setup() {
28     randomSeed(analogRead(A0));
29
30     pinMode(BUZZER, OUTPUT);
31     for (int i = 0; i < 4; i++)
32     {
33         pinMode(botoes[i], INPUT_PULLUP);
34         pinMode(leds[i], OUTPUT);
35     }
36     //Lógica invertida: LOW = pressionado
37     pinMode(MODO, INPUT_PULLUP);
38
39     //Se o botão MODO estiver pressionado
40     //o jogo entrará no modo de tempo progressivo.
41     escolherModo();
42     iniciarJogo();
43 }
44
45 void loop() {
46     mostrarSequencia();
47
48     if (!lerJogada())
49     {
50         somErro();
51         piscarTodosLeds(3);
52         delay(1000);
53         //Permite novo modo ao errar
54         escolherModo();
55         iniciarJogo();
56     }
57     else
58     {
59         nivel++;
60         delay(1000);
61     }
62 }
```

```

63
64 void escolherModo()
65 {
66     if (digitalRead(MODO) == LOW)
67     {
68         modo = 2;
69         tempoBase = 500;
70         //Feedback visual: piscadas duplas = modo progressivo
71         piscarTodosLeds(2);
72     }
73     else
74     {
75         modo = 1;
76         tempoBase = 500;
77         //Feedback visual: 1 piscada = modo fixo
78         piscarTodosLeds(1);
79     }
80 }
81
82 void iniciarJogo()
83 {
84     nivel = 1;
85     for (int i = 0; i < MAX SEQ; i++)
86     {
87         //Captura uma sequência aleatória
88         sequencia[i] = random(0, 4);
89     }
90 }
91
92 void mostrarSequencia()
93 {
94     int tempo = tempoBase;
95     //Se o modo 2 foi selecionado...
96     if (modo == 2)
97     {
98         //...o tempo passa a ser progressivo
99         tempo = max(150, tempoBase - (nivel * 20));
100    }
101
102    for (int i = 0; i < nivel; i++)
103    {
104        acenderLedComSom(sequencia[i]);
105        delay(tempo);
106        apagarLeds();
107        delay(tempo / 2);
108    }
109 }
110
111 //Verifica se a jogada está correta
112 bool lerJogada()
113 {
114     for (int i = 0; i < nivel; i++)
115     {
116         int entrada = esperarBotao();
117         if (entrada != sequencia[i])
118         {
119             return false;
120         }
121     }
122     return true;
123 }

```

```

124
125 //Espera apertar algum botão
126 int esperarBotao()
127 {
128     while (true)
129     {
130         for (int i = 0; i < 4; i++)
131         {
132             if (digitalRead(botoes[i]) == LOW)
133             {
134                 acenderLedComSom(i);
135                 delay(200);
136                 apagarLeds();
137                 while (digitalRead(botoes[i]) == LOW);
138                 return i;
139             }
140         }
141     }
142 }
143
144 //Acende o LED da sequência
145 void acenderLedComSom(int i)
146 {
147     digitalWrite(leds[i], HIGH);
148     tone(BUZZER, tons[i], 200);
149 }
150
151 //Apaga todos os LEDs
152 void apagarLeds()
153 {
154     for (int i = 0; i < 4; i++)
155     {
156         digitalWrite(leds[i], LOW);
157     }
158 }
159
160 //Pisca todos os LEDs
161 void piscarTodosLeds(int vezes)
162 {
163     for (int i = 0; i < vezes; i++)
164     {
165         for (int j = 0; j < 4; j++) digitalWrite(leds[j], HIGH);
166         delay(200);
167         for (int j = 0; j < 4; j++) digitalWrite(leds[j], LOW);
168         delay(200);
169     }
170 }
171
172 //Emite o som de erro
173 void somErro()
174 {
175     //Atva o som em 100Hz por 600ms
176     tone(BUZZER, 100, 600);
177     delay(600);
178     //Desativa o som
179     noTone(BUZZER);
180 }

```