



# Programação & Arduino

Desenvolvimento de Sistemas

## Jogo da Memória (Genius)

Neste projeto vamos apresentar o "Jogo da Memória" também chamado de "Genius", [jogo](#) de grande sucesso na década de 80. O [jogo](#) apresenta 4 botões e 4 leds que acenderão em uma sequência aleatória, com sons definidos para cada um dos leds. O jogador deverá clicar nos botões de acordo com a sequência gerada pela [Arduino](#). A cada jogada, a sequência é acrescida de um novo item. Ganha quem conseguir repetir a maior sequência possível. O jogo é limitado a 100 jogadas, mas você poderá aumentar se desejar.



### Materiais necessários:

- 1X Protoboard 830 pontos
- 1X Kit cabos ligação macho / macho
- 4X Led 5mm
- 4X Kit cabos ligação/ macho
- 4X Resistor de 150  $\Omega$  maior
- 4X Push button 6X6X5mm
- 1X Arduino UNO

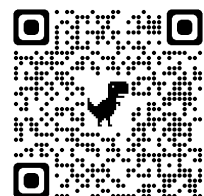
### Como o projeto deve funcionar.

Ao executar o programa, todos os leds acenderão e é dado um sinal sonoro.

Aleatoriamente, acenderá o primeiro led entre os 4 leds do [jogo](#). O programa irá aguardar que o jogador clique no botão correspondente. Cada led aceso tem um som correspondente. Se o jogador acertar o botão, repete-se a sequência com o acréscimo aleatório do segundo led. Se o jogador errar, acenderá um sinal sonoro indicando o fim do [jogo](#). Novo led se acenderá....

Ganha o jogador que acertar mais jogadas, ou seja, memorizar a maior sequência de sons.

Projeto disponível em: <https://github.com/2PAC-DO-NORDESTE/Jogo-da-Memoria..git>



```

/*****
**
**
**
**  Jogo 3 - Jogo Genius
**  Criado por: https://meetarduino.wordpress.com
**  Adaptado por: Angelo Luis Ferreira
**
**  http://squids.com.br/arduino
**
** ****
** /

```

```

//Vamos começar definindo as notas para os sons
#define NOTE_D4 294
#define NOTE_G4 392
#define NOTE_A4 440
#define NOTE_A5 880

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x20, 16, 2);

// criando o array para os 4 sons para sortear um som
int tons[4] = { NOTE_A5, NOTE_A4, NOTE_G4, NOTE_D4 };
// Nossa sequência de até 100 itens vai começar vazia.
int sequencia[100] = {};
// Indica a rodada atual que o jogo se encontra.
int rodada_atual = 0;
// Indica o passo atual dentro da sequência, é usado enquanto a sequência
// está sendo reproduzida.
int passo_atual_na_sequencia = 0;
int melhor_sequencia = 0;
/*
 * Indica o pino de áudio, leds e botões.
 * Os pinos de leds e botões estão em ordem, relacionados uns aos outros, ou
 * seja, o primeiro led está relacionado ao primeiro botão. A ordem destas
 * sequências também estão relacionadas a ordem dos tons.
 */
int pinoAudio = 12;
int pinosLeds[4] = { 2, 4, 6, 8 };
int pinosBotoes[4] = { 3, 5, 7, 9 };

// Indica se um botão foi pressionado durante o loop principal.
int botao_pressionado = 0;
// Flag indicando se o jogo acabou.
int perdeu_o_jogo = false;

void setup() {
    lcd.init(); // inicializa LCD
    lcd.backlight(); // ativa led de backlight
    lcd.setCursor(0, 0); // selecionando coluna 0 e linha 0
    lcd.print("Jogo Genius");
    lcd.setCursor(0, 1);
    lcd.print("Rodada Atual: ");

    // Definindo o modo dos pinos dos Leds como saída.
    for (int i = 0; i <= 3; i++) {
        pinMode(pinosLeds[i], OUTPUT);
    }

    // Definindo o modo dos pinos dos Botões como pullup interno.
    for (int i = 0; i <= 3; i++) {
        pinMode(pinosBotoes[i], INPUT_PULLUP);
    }

    // Definindo o modo do pino de Áudio como saída.

```

```

pinMode(pinoAudio, OUTPUT);

// Inicializando o random através de uma leitura da porta analógica.
// Esta leitura gera um valor variável entre 0 e 1023.
randomSeed(analogRead(A0));
}

void loop() {
// Se perdeu o jogo reinicializamos todas as variáveis.
if (perdeu_o_jogo) {
    int sequencia[100] = {};
    rodada_atual = 0;
    passo_atual_na_sequencia = 0;
    perdeu_o_jogo = false;
    lcd.backlight();
    lcd.clear();
    lcd.setCursor(0, 0);           // selecionando coluna 0 e linha 0
    lcd.print("Jogo Genius");
    lcd.setCursor(0, 1);
    lcd.print("Rodada Atual: ");
}

// Toca um som de início para anunciar que o jogo está começando quando é a
primeira rodada.
if (rodada_atual == 0) {
    tocarSomDeInicio();
    delay(500);
}

// Chama a função que inicializa a próxima rodada.
proximaRodada();
// Reproduz a sequência atual.
reproduzirSequencia();
// Aguarda os botões serem pressionados pelo jogador.
aguardarJogador();

// Aguarda 1 segundo entre cada jogada.
delay(1000);
}

// Sorteia um novo item e adiciona na sequência.
void proximaRodada() {
    lcd.setCursor(13, 1);

    int numero_sorteado = random(0, 4);
    sequencia[rodada_atual++] = numero_sorteado;
    lcd.print(rodada_atual);
}

// Reproduz a sequência para ser memorizada.
void reproduzirSequencia() {
    for (int i = 0; i < rodada_atual; i++) {
        tone(pinoAudio, tons[sequencia[i]]);
    }
}

```

```

    digitalWrite(pinosLeds[sequencia[i]], HIGH);
    delay(500);
    noTone(pinoAudio);
    digitalWrite(pinosLeds[sequencia[i]], LOW);
    delay(100);
}
noTone(pinoAudio);
}

// Aguarda o jogador iniciar sua jogada.
void aguardarJogador() {
    for (int i = 0; i < rodada_atual; i++) {
        aguardarJogada();

        // verifica a jogada
        if (sequencia[passo_atual_na_sequencia] != botao_pressionado) {
            gameOver(); // perdeu
        }

        // para o jogo se perdeu
        if (perdeu_o_jogo) {
            break;
        }
        passo_atual_na_sequencia++;
    }

    // Redefine a variável para 0.
    passo_atual_na_sequencia = 0;
}

void aguardarJogada() {
    boolean jogada_efetuada = false;
    while (!jogada_efetuada) {
        for (int i = 0; i <= 3; i++) {
            if (!digitalRead(pinosBotoes[i])) {
                // Dizendo qual foi o botao pressionado.
                botao_pressionado = i;

                tone(pinoAudio, tons[i]);
                digitalWrite(pinosLeds[i], HIGH);
                delay(300);
                digitalWrite(pinosLeds[i], LOW);
                noTone(pinoAudio);

                jogada_efetuada = true;
            }
        }
        delay(10);
    }
}

void gameOver() {
    // GAME OVER.
    lcd.clear();
}

```

```
    lcd.setCursor(0, 0); // selecionando coluna 0 e linha 0
    lcd.print("Voce perdeu! :(");

    for (int i = 0; i <= 3; i++) {
        tone(pinoAudio, tons[i]);
        digitalWrite(pinosLeds[i], HIGH);
        delay(200);
        digitalWrite(pinosLeds[i], LOW);
        noTone(pinoAudio);
    }

    tone(pinoAudio, tons[3]);
    for (int i = 0; i <= 3; i++) {
        digitalWrite(pinosLeds[0], HIGH);
        digitalWrite(pinosLeds[1], HIGH);
        digitalWrite(pinosLeds[2], HIGH);
        digitalWrite(pinosLeds[3], HIGH);
        lcd.backlight();
        delay(100);
        digitalWrite(pinosLeds[0], LOW);
        digitalWrite(pinosLeds[1], LOW);
        digitalWrite(pinosLeds[2], LOW);
        digitalWrite(pinosLeds[3], LOW);
        lcd.noBacklight();
        delay(100);
    }
    noTone(pinoAudio);

    perdeu_o_jogo = true;
}

void tocarSomDeInicio() {
    tone(pinoAudio, tons[0]);
    digitalWrite(pinosLeds[0], HIGH);
    digitalWrite(pinosLeds[1], HIGH);
    digitalWrite(pinosLeds[2], HIGH);
    digitalWrite(pinosLeds[3], HIGH);
    delay(500);
    digitalWrite(pinosLeds[0], LOW);
    digitalWrite(pinosLeds[1], LOW);
    digitalWrite(pinosLeds[2], LOW);
    digitalWrite(pinosLeds[3], LOW);
    delay(500);
    noTone(pinoAudio);
}
```