**Documentação**

**Aluno:** João Pedro Sales de Deus

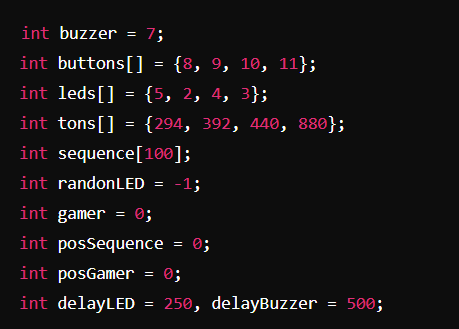
**Professor:** João Paulo Aramuni

**Montagem:**

1. **Placa Arduino Uno**: É o cérebro do projeto, responsável por controlar os componentes conectados a ela.
2. **Protoboard (Breadboard)**: Uma placa de ensaio onde os componentes são inseridos e conectados através de fios condutores.
3. **LEDs**: Dois LEDs (um vermelho e um azul) estão presentes no circuito. Eles podem ser usados para indicar o status do sistema ou para alertar sobre as leituras do sensor.
4. **Buzzer**: Pode ser utilizado para emitir alertas sonoros ou para indicar eventos específicos, como uma mudança nas condições ambientais detectadas pelo sensor DHT11.
5. **Fios Condutores**: Os fios são usados para fazer as conexões elétricas entre os componentes na protoboard e a placa Arduino.
6. **Botões: Botões para o usuário poder pressionar para começar o jogo.**

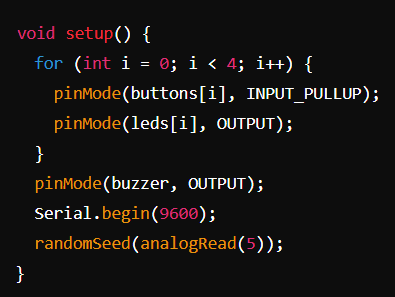
**Explicação do Código em C++:**

**Declaração de Variáveis:**



1. **buzzer**: Pin conectado ao buzzer que emitirá sons.
2. **buttons**: Array contendo os pinos dos botões.
3. **leds**: Array contendo os pinos dos LEDs.
4. **tons**: Array com as frequências dos tons que serão emitidos pelo buzzer.
5. **sequence**: Array que armazenará a sequência gerada pelo jogo.
6. **randonLED**: Variável que armazenará o LED sorteado na sequência.
7. **gamer**: Variável que controla o modo do jogo (0 para modo Genius, 1 para modo Jogador).
8. **posSequence**: Armazena a última posição da sequência de LEDs.
9. **posGamer**: Posição das sequências de botões que o jogador pressiona.
10. **delayLED** e **delayBuzzer**: Delays usados para controlar a duração dos LEDs e tons.

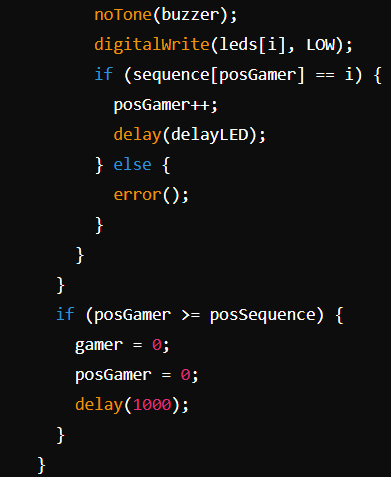
**Função setup:**



1. Configura os pinos dos botões como entradas com resistores pull-up internos.
2. Configura os pinos dos LEDs e do buzzer como saídas.
3. Inicializa a comunicação serial.
4. Semeia o gerador de números aleatórios usando a leitura de um pino analógico não utilizado.

**Função loop:**





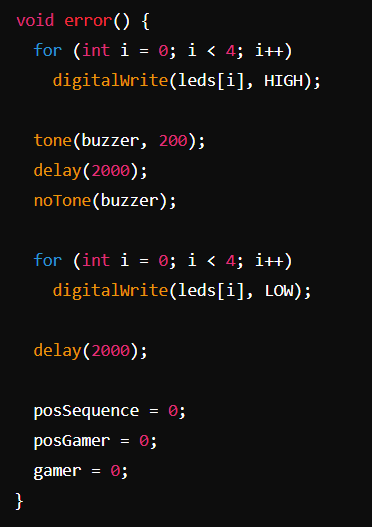
**Modo Genius (gamer == 0)**:

1. Gera um valor aleatório entre 0 e 3.
2. Adiciona esse valor à sequência.
3. Acende os LEDs e toca os tons na ordem da sequência.
4. Incrementa a posição da sequência e muda para o modo jogador.

**Modo Jogador (gamer == 1)**:

1. Espera o jogador pressionar os botões.
2. Verifica se o botão pressionado corresponde à sequência gerada.
3. Se o jogador erra, chama a função error().
4. Se o jogador acerta a sequência, volta ao modo Genius para gerar a próxima sequência.

**Função error:**



1. Acende todos os LEDs.
2. Emite um tom de erro no buzzer.
3. Desliga todos os LEDs após um delay.
4. Reseta as variáveis para reiniciar o jogo.

**Resultados Obtidos:**

Este código implementa um jogo Genius no Arduino, onde o microcontrolador gera uma sequência de LEDs e tons que o jogador deve reproduzir corretamente. Se o jogador errar a sequência, o jogo exibe um erro visual e sonoro e reinicia a sequência.