

## Sessão 05

### Pedra, papel, tesoura, lagarto, Spock no arduino

Professor Thiago Goveia

### Objetivos da sessão:

- Aplicar a lógica de programação desenvolvida em C em um ambiente de hardware com Arduino.
- Controlar entradas digitais (teclado matricial) e saídas digitais (servo e LEDs) para criar uma interface de jogo física.
- Avaliar o processo de aprendizagem do primeiro módulo

### Materiais:

- [Repositório](#) do projeto criado na Sessão 01 no GitHub.
- Kit Arduino (Placa UNO, Protoboard, Fios Jumper).
- 1 teclado matricial 4x4.
- 6 LEDs e 6 Resistores (220Ω).
- 1 Servomotor.

### Sumário:

<a href="#">1. Introdução: Por que um protótipo Físico?</a>	<a href="#">2</a>
<a href="#">2. A Caixa de Ferramentas do Arduino</a>	<a href="#">2</a>
<a href="#">3. Mão na Massa!</a>	<a href="#">2</a>
<a href="#">3.1 Implementando a Lógica do PPT no loop()</a>	<a href="#">2</a>
<a href="#">3.2 Evoluindo para PPTLS</a>	<a href="#">3</a>
<a href="#">4. Momento de confronto e avaliação.</a>	<a href="#">3</a>



## Conteúdo da sessão:

### 1. Introdução: Por que um protótipo Físico?

Nas sessões anteriores, criamos a lógica completa do nosso jogo em linguagem C. Hoje, vamos dar vida a essa lógica! O objetivo é provar que um bom algoritmo é universal: a mesma lógica que funciona em um terminal de computador pode controlar luzes, botões e motores no mundo real.

Vamos construir um protótipo de um console de jogo utilizando componentes eletrônicos para criar uma experiência interativa e tátil.

### 2. A Caixa de Ferramentas do Arduino

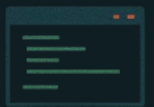
- **Teclado Matricial (Entrada):** Serão as nossas teclas de comando. Cada botão representará uma jogada (Pedra, Papel, etc.). É uma entrada digital simples: ou está pressionado (HIGH) ou não está (LOW). Utilizaremos a biblioteca `Keypad.h` para facilitar a manipulação dos dados do teclado.
- **LEDs (Saída):** Serão nosso feedback visual. Eles mostrarão os pontos dos jogadores. Para ligar um LED, basta usar o comando `digitalWrite(pino, HIGH);`.
- **Servomotor (Saída):** Será a "mão" do nosso oponente. Diferente de um motor comum, ele não gira continuamente; ele se move para um ângulo específico (geralmente entre 0 e 180 graus) e permanece lá. Para controlá-lo, usamos a biblioteca `Servo.h` e o comando `servo.write(ângulo)`.

### 3. Mão na Massa!

#### 3.1 Implementando a Lógica do PPT no `loop()`

Use o código-base do arquivo `sessao_05_ppt.ino`, disponível no [Tinkercad](#) e na pasta da Sessão 05, e complete a função `loop()`.

Quando o jogador pressionar uma tecla (de 1 a 3), deve ser sorteado um número para a jogada do computador e o servomotor deve ser movido



para um dos ângulos correspondentes a pedra ( $180^\circ$ ), papel ( $90^\circ$ ) ou tesoura ( $0^\circ$ ).

**Dica:** Antes de sortear um número, crie o seguinte algoritmo de teste: Se você pressionar 1, o servo deve ser movimentado para “pedra”, se 2, para papel, se 3, para tesoura.

Utilize os LEDs (em série com os resistores) para marcar os pontos. quem completar 4 pontos primeiro, ganha e a contagem é reiniciada.

Após concluir a simulação, faça a montagem do circuito no arduino e teste!

**Não se esqueça de salvar o arquivo final .ino, no github!**

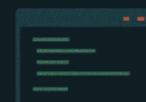
### 3.2 Evoluindo para PPTLS

Nosso jogo de 3 opções funciona. Mas e se quisermos adicionar Lagarto e Spock? Se usarmos uma cadeia de `if/else`, o código ficará enorme e confuso. É aqui que a **matriz de regras** que aprendemos na aula de C se torna uma ferramenta poderosa!

Modifique seu algoritmo para contemplar as regras “Lagarto” e “Spock”. Após as modificações e testes, salve-o no Github.

## 4. Momento de confronto e avaliação.

Avalie a implementação de outro grupo. O que pode ser melhorado? O que fizeram melhor que você?



**SPOILER  
ALERT**

**Próxima sessão: Jogo da forca**

