



Introdução a Eletônica básica (IoT)

Por: Marcos Augusto Campagnaro Mucelini



Nesta aula veremos os conceitos de:

- **Botões**
 - Push
 - Tátil
- **Chaves**
 - Switch
 - Slide
- **Praticas**





O que são botões e chave

Embora os termos ‘botão’ e ‘chave’ sejam frequentemente usados como sinônimos no dia a dia da eletrônica, e ambos possam parecer semelhantes à primeira vista, há diferenças fundamentais entre esses dois componentes. Vamos tentar entender essas diferenças e porque são importantes na hora de projetar circuitos para os nossos projetos.





Botões

Botões (*Push buttons*)

Características:

- São dispositivos **momentâneos**, ou seja, só **permanecem ativos** enquanto estão sendo **pressionados**.
- Retornam automaticamente ao seu estado original após serem soltos.
- A maioria dos botões simples é do tipo SPST momentâneo (veremos posteriormente o que isso significa) normalmente aberto (NO).





Botões (*Push buttons*)

Comportamento:

- Geram pulso curto (**HIGH** ou **LOW**) ao serem pressionados.
- Requerem geralmente tratamento de debounce no software ou hardware, pois o contato pode oscilar por alguns milissegundos ao pressionar/soltar.





Botões (*Push buttons*)

Aplicações:

- Entrada de comandos do usuário (ex: iniciar, parar, resetar, ligar)
- Contadores, controle de estado



Botão (Push up)



Botão (Táctil)





Chaves (*Switchs*)

Chaves (*Switch*)

Características:


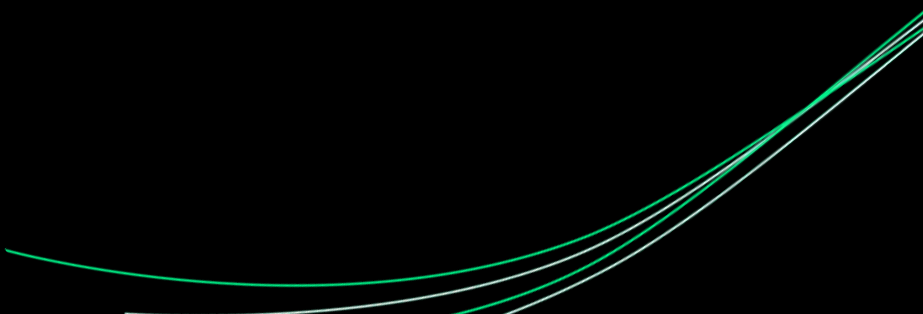
- São **dispositivos de posição**, ou seja, **mantêm o estado** em que foram colocados **até serem alternados novamente**.
- Podem ter duas ou mais posições fixas (como ON-OFF ou ON-OFF-ON). Na grande maioria são do tipo SPDT (veremos o que isso significa mais tarde).
- **Existem vários tipos**: chave gangorra, toggle, slide, rotativa, fim de curso, etc.





Chaves (*Switch*)

Comportamento:

- Geram um **estado estável** (**HIGH** ou **LOW**) que pode ser lido continuamente.
 - **Não precisam de debounce** se usados com leitura esporádica, embora seja recomendado em leitura contínua.
- 
- 



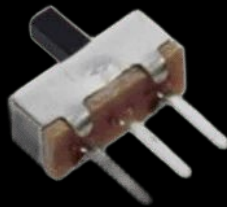
Chaves (*Switch*)

Aplicações:

- Liga/desliga de circuitos
- Seleção de modos (ex: manual/automático)
- Comutação de estados em tempo real



Chave Slide

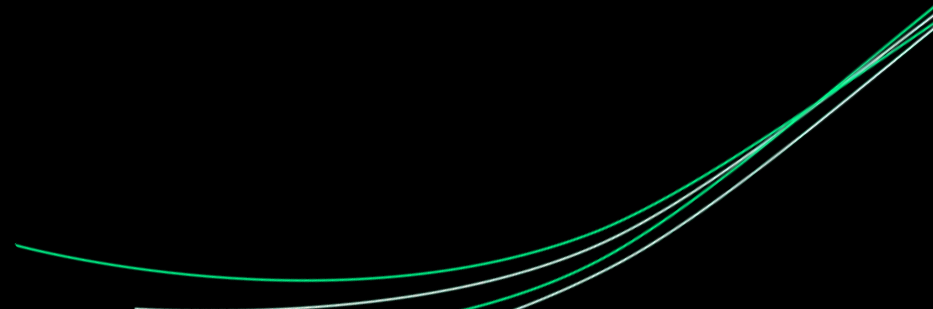


Chave Toggle (Alavanca)





Atividades





Atividade

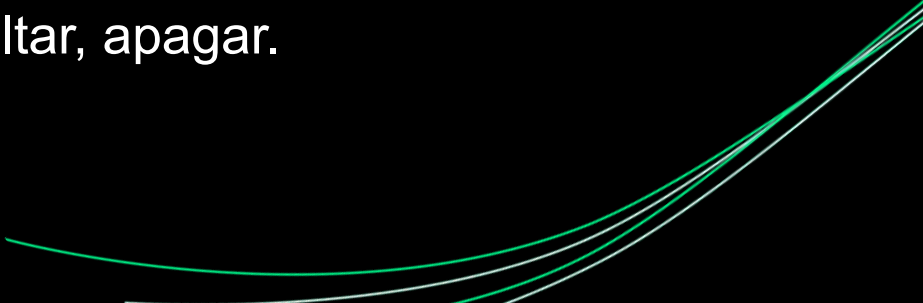
Acender LED com botão (circuito simples)

Objetivo: entender como o botão funciona como interruptor.

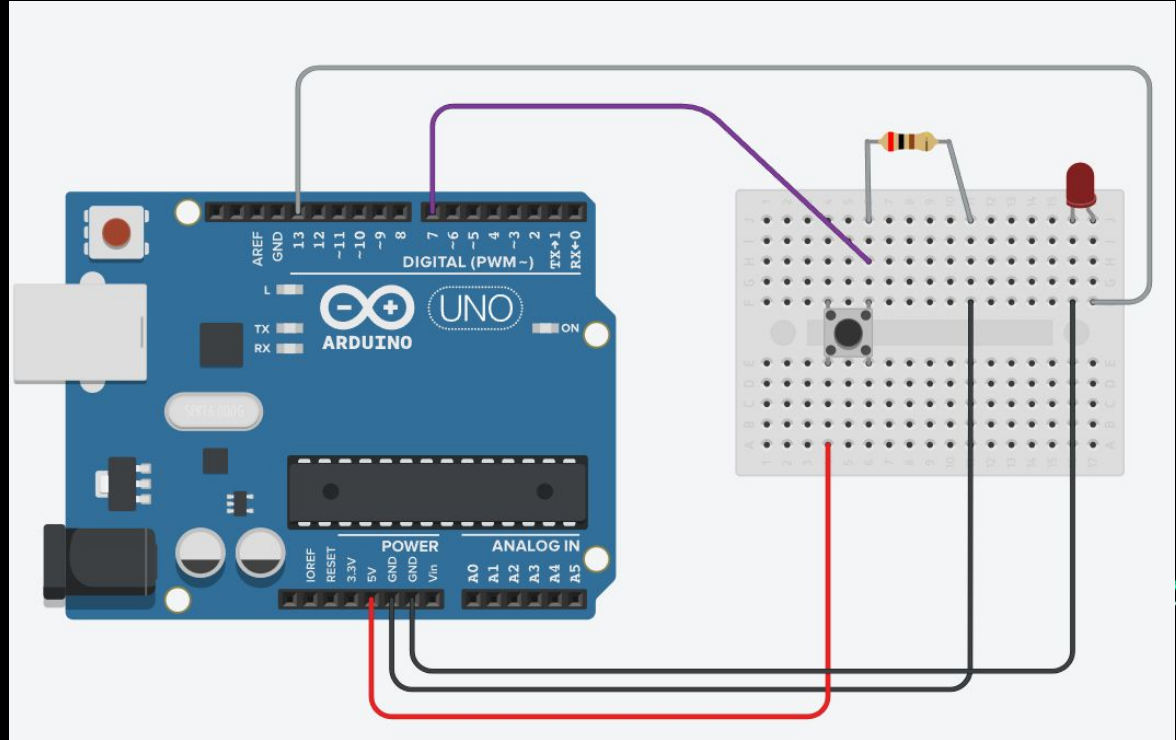


Componentes: 1 LED, 1 botão, resistores (para o LED e botão), protoboard, fios, Arduino.

Tarefa: Ao pressionar o botão, o LED deve acender; ao soltar, apagar.



Diagrama



Programação



no início

definir pino 13 como BAIXO

definir pino 7 como BAIXO

para sempre

definir botao como ler pino digital 7

se botao = ALTO, então

definir pino 13 como ALTO

aguardar 2 s

outro

definir pino 13 como BAIXO

aguardar 1 s



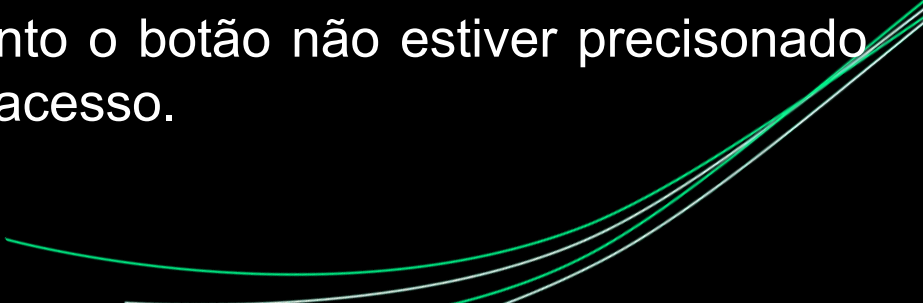
Atividade

Acender LED com botão inverso (circuito simples)

Objetivo: Entender como a lógica de programação afeta o resultado obtido.

Componentes: 1 LED, 1 botão, resistores (para o LED e botão), protoboard, fios, Arduino.

Tarefa: Enquanto o botão não estiver precisonado manter o LED acesso.



Programação



no início

definir pino 13 como BAIXO

definir pino 7 como BAIXO

para sempre

definir botao como ler pino digital 7

se botao = BAIXO, então

definir pino 13 como ALTO

aguardar 2 s

outro

definir pino 13 como BAIXO

aguardar 1 s



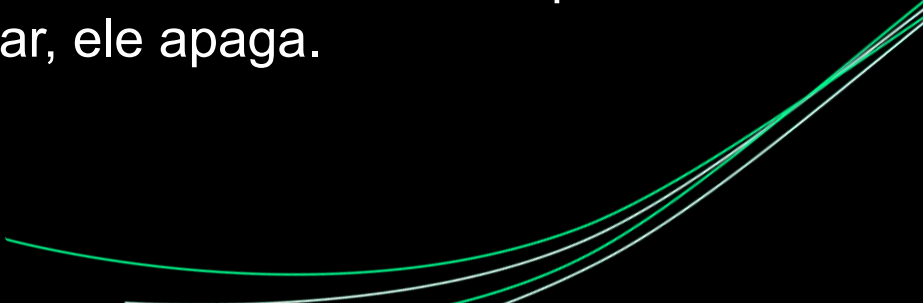
Atividade

LED pisca enquanto o botão é pressionado
(circuito simples)

Objetivo: trabalhar com loops condicionais.

Componentes: 1 LED, 1 botão, resistores (para o LED e botão), protoboard, fios, Arduino.

Tarefa: Pressionar o botão faz o LED piscar a cada 500ms. Ao soltar, ele apaga.



Programação



no início

definir pino 13 como BAIXO

definir pino 7 como BAIXO

para sempre

definir botao como ler pino digital 7

repetir enquanto botao = ALTO

definir pino 13 como ALTO

aguardar 0.5 s

definir pino 13 como BAIXO

aguardar 0.5 s

definir botao como ler pino digital 7



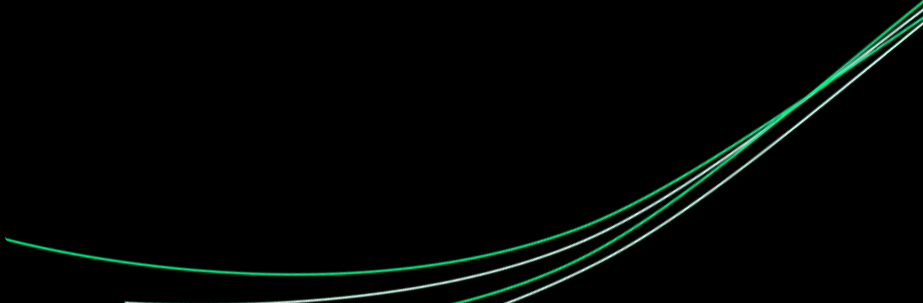
Atividade

Dois LEDs e um botão – alternância

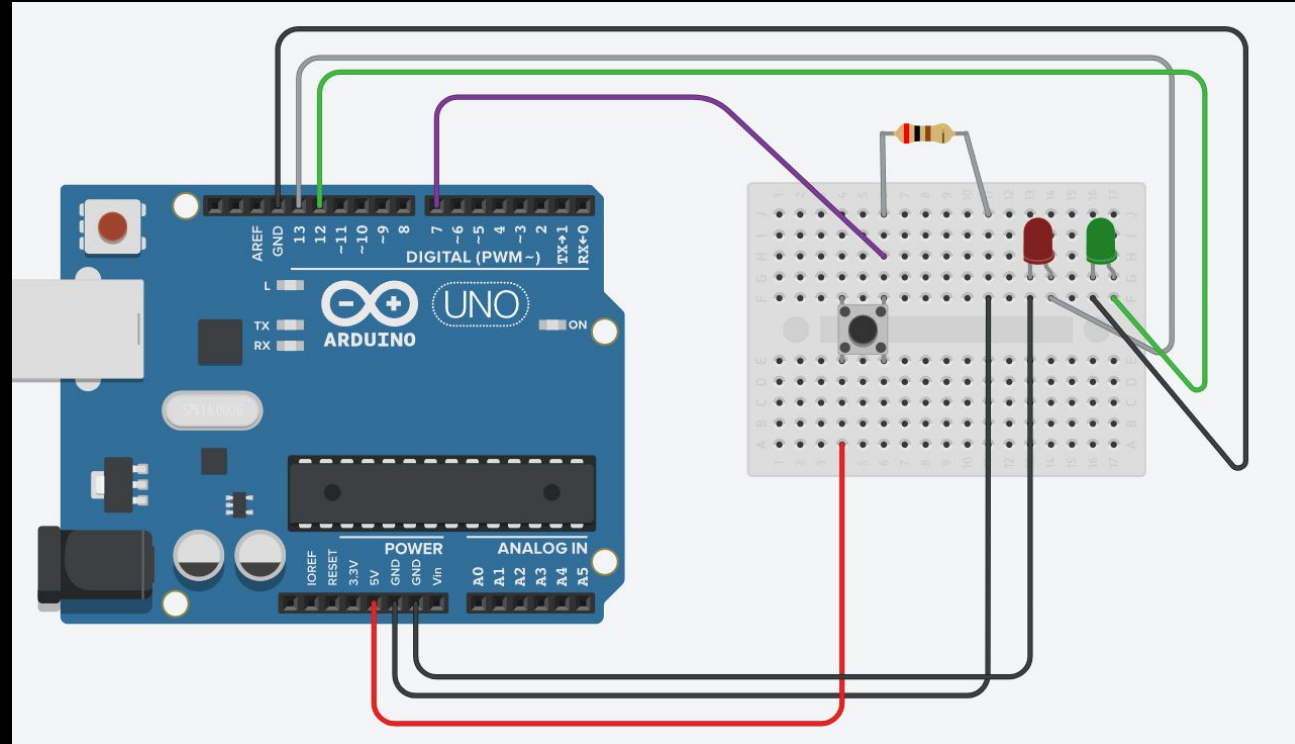
Objetivo: usar variável de estado.

Componentes: 2 LED, 1 botão, resistores (para o LED e botão), protoboard, fios, Arduino.

Tarefa: Cada vez que o botão é pressionado, um LED apaga e o outro acende.



Diagrama



Programação



no início

definir pino 13 como BAIXO
definir pino 7 como BAIXO

para sempre

definir botao como ler pino digital 7
se botao = ALTO, então
 definir pino 13 como ALTO
 definir pino 12 como BAIXO
outro
 definir pino 13 como BAIXO
 definir pino 12 como ALTO



Atividade


“Sequência de LEDs” com botão

Objetivo: criar efeito de “corrida de luz” (tipo Knight Rider).

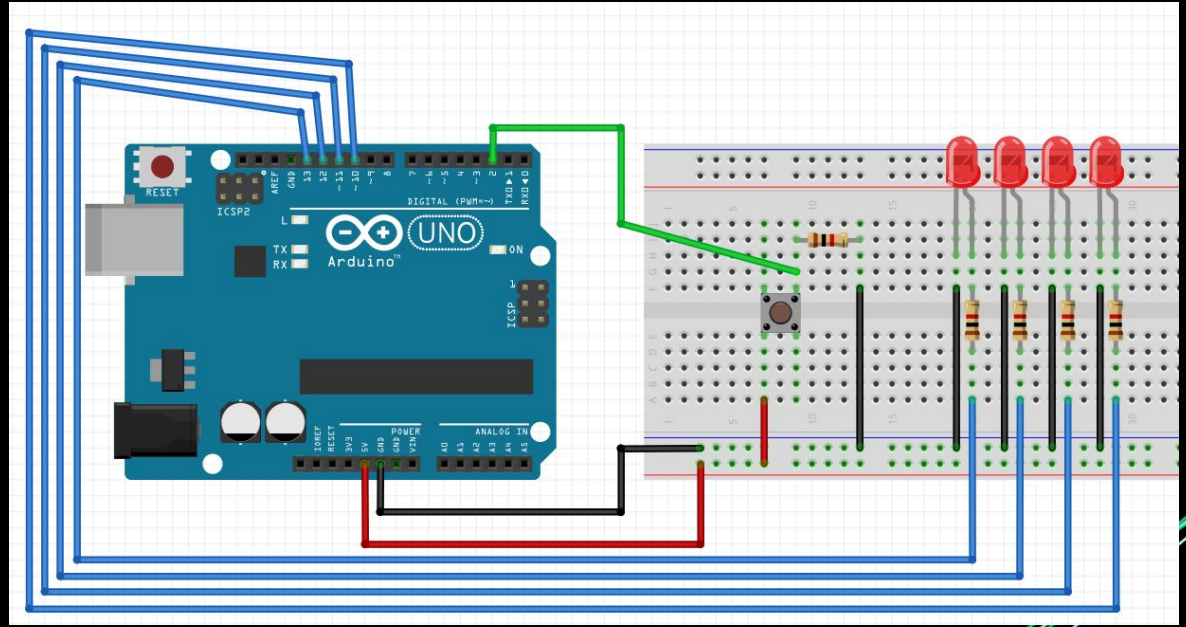


Componentes: 4 ou mais LEDs, 1 botão, resistores (para o LED e botão), protoboard, fios, Arduino.

Tarefa: Ao segurar o botão, LEDs fazem animação para esquerda e direita.

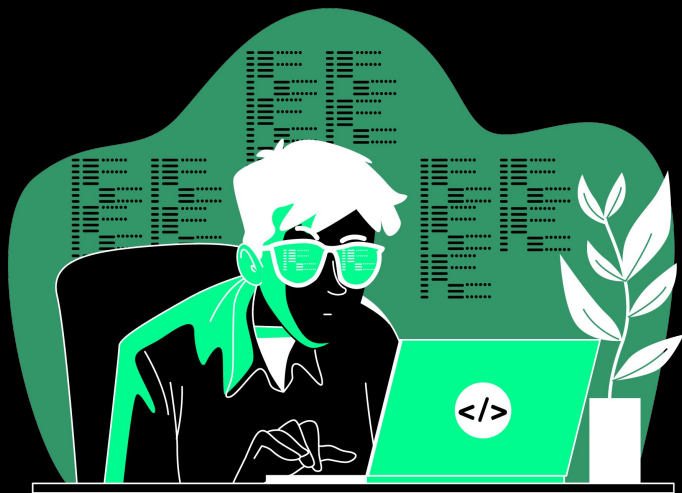


Diagrama



Programação





Obrigado!