



Controlo programado com Arduino

Prática 1: O Meu Primeiro LED (Ligado Contínuo)

A primeira paragem da nossa viagem consiste em entender como flui a energia pela nossa placa Arduino Uno e aprender a dar instruções muito básicas. Hoje conseguirás acender um componente emissor de luz (LED) programando as tuas primeiras linhas de código em C++.

Objetivos de Aprendizagem

Ao finalizar esta sessão, serás capaz de:

- Identificar a estrutura fundamental de um programa em Arduino (`setup` e `loop`).
- Configurar um pino digital como canal de saída utilizando `pinMode` .
- Controlar a passagem de corrente elétrica através do comando `digitalWrite` .
- Ligar corretamente um LED e uma resistência protetora numa placa de ensaio (Protoboard).

Componentes Necessários (TinkerCad)

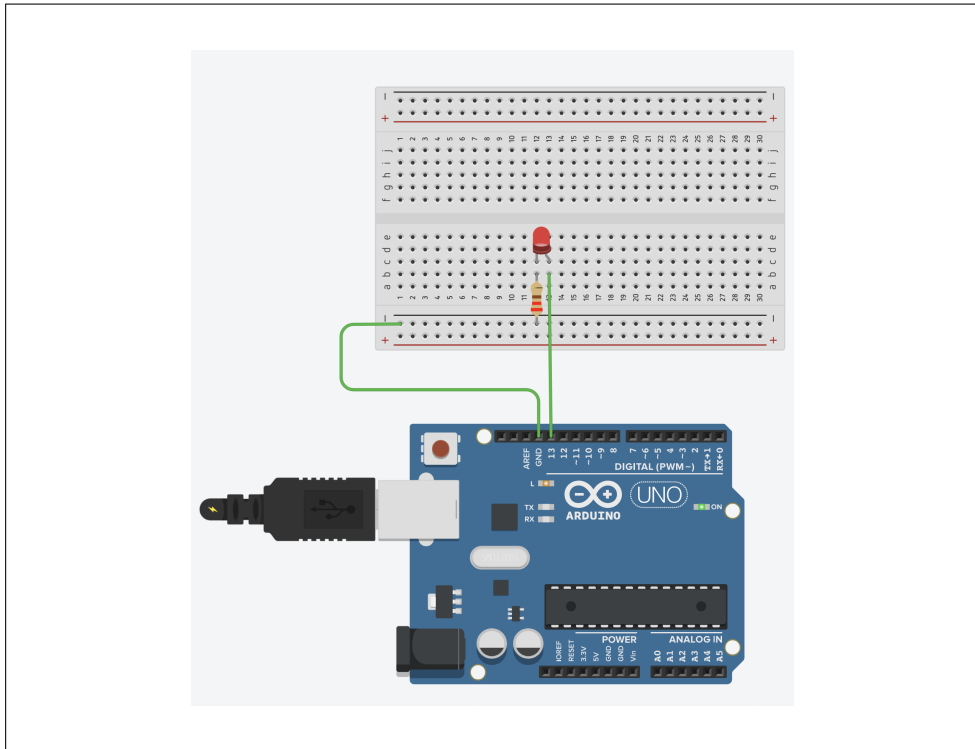
Para esta prática apenas precisarás de arrastar os seguintes componentes no teu ecrã:

- 1 Placa Arduino Uno.
- 1 Placa de ensaio pequena (Protoboard).
- 1 Díodo LED (podes escolher a cor que mais gostares).
- 1 Resistência de $220\ \Omega$ (*importante! Se não colocares a resistência, o LED poderá queimar devido ao excesso de corrente*).
- Cabos de ligação virtuais.

□ Esquema de Montagem na Protoboard

Para ligar os componentes no TinkerCad de forma segura, segue atentamente estas diretrizes:

Configuração Inicial do objeto Terminal



□ Código Base: A Estrutura

Abre o editor de código no TinkerCad, seleciona o modo **Texto** (apaga o código que vem por defeito se for necessário) e escreve o seguinte programa. Lê atentamente os comentários em cor verde para compreender o que faz cada linha.

```
1 // Pratica 1: Ligado contínuo de um LED
2 // Esta função é executada uma única vez quando o Arduino se liga ou se reinicia
3 void setup() {
4     // Configura o Pino Digital 13 como uma SAÍDA (OUTPUT) de corrente
5     pinMode(13, OUTPUT);
6 }
7
8 // Esta função é executada constantemente de forma infinita (em loop)
9 void loop() {
10    // Envia um estado ALTO (HIGH / 5 Volts) ao Pino 13 para acender o LED
11    digitalWrite(13, HIGH);
12 }
```

Listing 1: Código para acender um LED continuamente.

□ Como funciona o código?


- `setup` : É o bloco de preparação. Aqui dizemos ao Arduino que pinos vamos usar e como se vão comportar (como entradas ou saídas de informação).
- `loop` : É o motor principal do programa. Quando termina de executar a última instrução deste bloco, volta imediatamente a começar a partir da primeira, num ciclo sem fim.
- `pinMode` : Define se o pino selecionado funciona a enviar energia (**OUTPUT**) ou a recebê-la ().
- `digitalWrite` : Permite aplicar tensão elétrica ao pino especificado. Se enviarmos **HIGH**, o pino fornecerá 5V. Se enviarmos **LOW**, ficará com 0V (sem corrente).

□ O Desafio da Prática 1

Agora que verificaste que o circuito funciona no simulador, demonstra o que aprendeste resolvendo este desafio:

1. Modifica as ligações no TinkerCad para que o LED não use o pino 13, mas sim o **pino digital 8**.
2. Realiza as alterações necessárias no código para que o programa continue a acender o LED de forma contínua.
3. *Pergunta para refletir:* Se quisesses apagar completamente o LED utilizando o programa, que instrução terias de modificar na função `loop` e com que parâmetro?

Experimenta mudar o limite do loop para 100 e verás o quão rápido o teu computador conta!

Este documento é publicado sob licença
 Creative Commons Atribuição 4.0 Internacional (CC BY 4.0)