

# Sistemas Reativos

Projeto Final da Disciplina

# Introdução



## O QUE É O GENIUS?

Genius é um jogo que ganhou popularidade durante a década de 1980. Seu principal objetivo é o estímulo à memorização através da combinação de cores e sons.

## COMO ELE FUNCIONA?

De maneira geral o jogo consiste em botões coloridos, iluminados por leds de 4 cores distintas que emitem sons harmônicos e acendem em uma sequência aleatória, a qual o jogador deve memorizar e repetir. Como toda sequência é gerada de maneira randômica, a probabilidade de que uma cadeia se repita é extremamente baixa e é esse efeito que garante o estímulo à memorização.

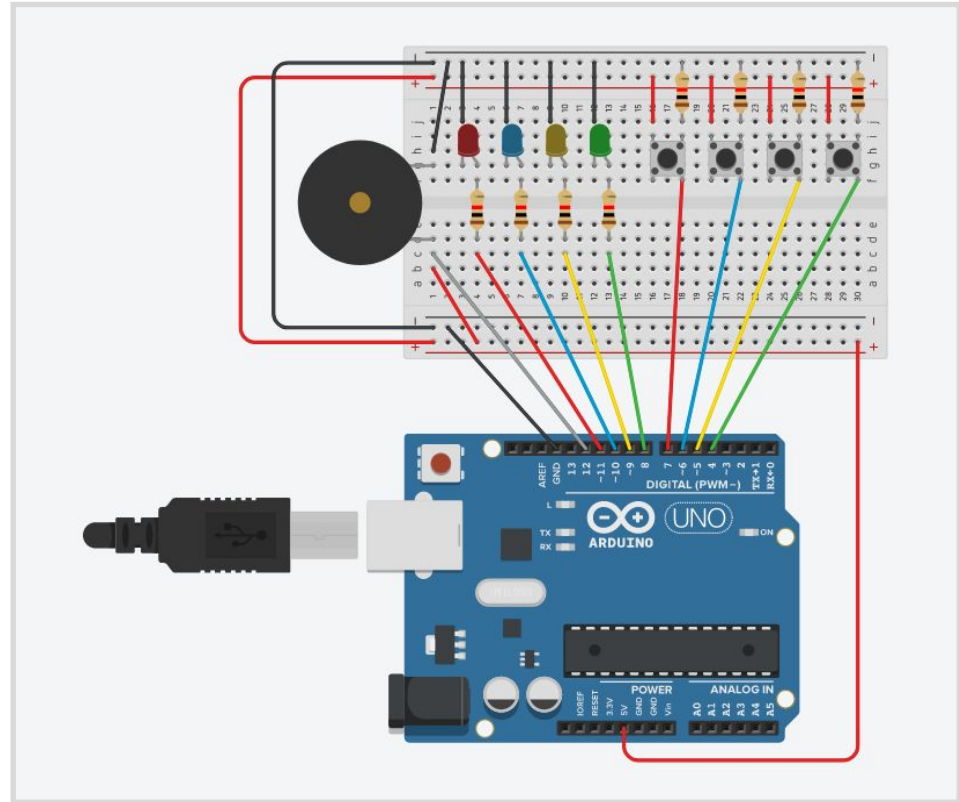
# MATERIAL UTILIZADO NO PROJETO

- 4 LEDs sendo cada um com seguintes cores: vermelho, azul, amarelo e verde;
- 8 Resistores de 220 ohm;
- 4 Botões do tipo push-button
- 1 Buzzer;
- 1 Protoboard (placa de prototipação);
- 1 Placa de Arduino UNO;
- 1 Cabo USB compatível com Arduino UNO;
- Fios jumper macho-macho;



# Modelo Físico

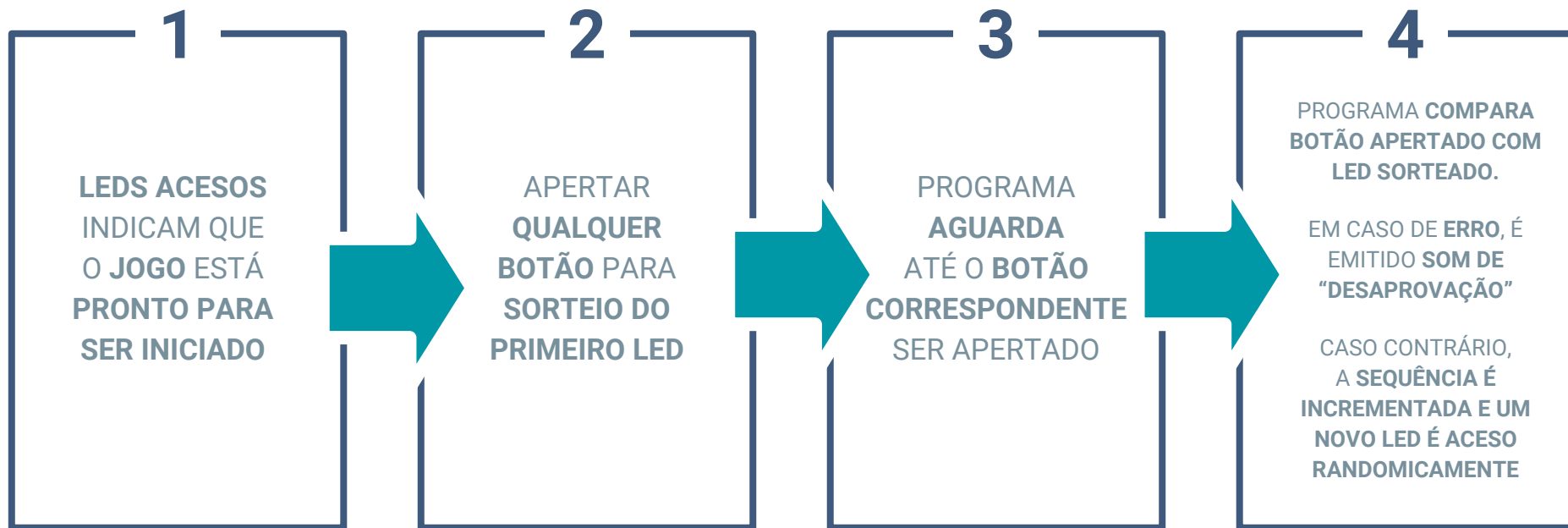
Montagem do circuito eletrônico na placa de Arduino



Como funciona o  
**programa?**

{.ino}

O programa descreve uma **sequência incremental e aleatória** para acendimento dos LEDs.  
O jogador deve segui-la utilizando os **botões correspondentes**.



Esse processo se repete até que ocorra  
um erro na entrada do usuário.

# Demonstração Prática

Entenda como funciona...



assista ao vídeo

<https://github.com/brunocriscuolo/reativos/blob/master/video-projeto-final.md>

Explorando o  
**código-fonte...**

**{.ino}**





## OBSERVAÇÃO

Toda a lógica do jogo está presente na função “**loop**( )” e as configurações em “**setup**( )”. Foram utilizadas algumas variáveis globais no desenvolvimento e toda a pinagem foi definida no início do programa através da declaração “**#define**”.

### **void** waitStart( ) { ... }

Essa é a função responsável pelo controle do início do jogo. Através dela o programa sinaliza que está pronto para ser iniciado, acendendo todos os LEDs e os apagando logo que qualquer um dos 4 botões é acionado.

### **int** input( ) { ... }

Função responsável por capturar as entradas entradas do usuário. Ela faz a verificação de qual botão foi acionado, retornando o valor inteiro do pino correspondente ao LED.

### **void** blinkLed( **int** ledPin ) { ... }

Função que permite os LEDs piscarem. Ela é responsável por acender e apagar os LEDs em um intervalo de tempo de 500ms.

### **void** toneLed( **int** ledPin ) { ... }

Função que atribui um tom ao Buzzer de acordo com o LED que estiver aceso. Para cada cor, existe um som correspondente que auxilia na memorização da sequência no jogo. Ela também determina o intervalo de 200ms para emissão desses sons.

### **void** error( ) { ... }

Essa função é responsável por emitir o som de “desaprovação” e piscar ininterruptamente o **LED vermelho**, indicado ao jogador que a sequência não foi reproduzida da maneira correta.

# Obrigado!



Todo conteúdo do projeto está disponível no GitHub

- Acesse o código-fonte: [https://github.com/brunocriscuolo/reativos/blob/master/projeto\\_final.ino](https://github.com/brunocriscuolo/reativos/blob/master/projeto_final.ino)
- Acesse a documentação: [https://github.com/brunocriscuolo/reativos/blob/master/projeto\\_final\\_doc.md](https://github.com/brunocriscuolo/reativos/blob/master/projeto_final_doc.md)