



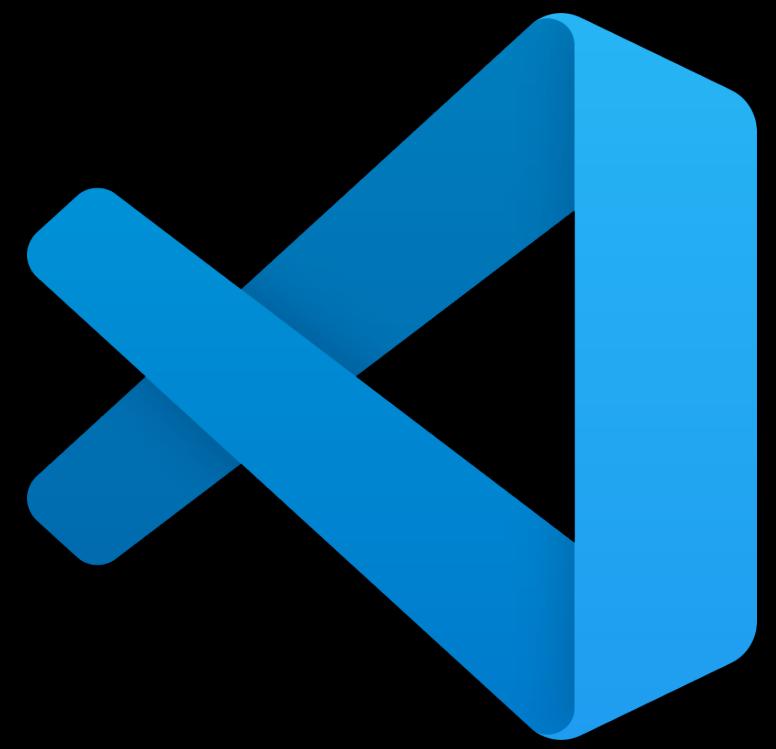
# Introdução

Revisão de Microcontroladores – Prática

Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

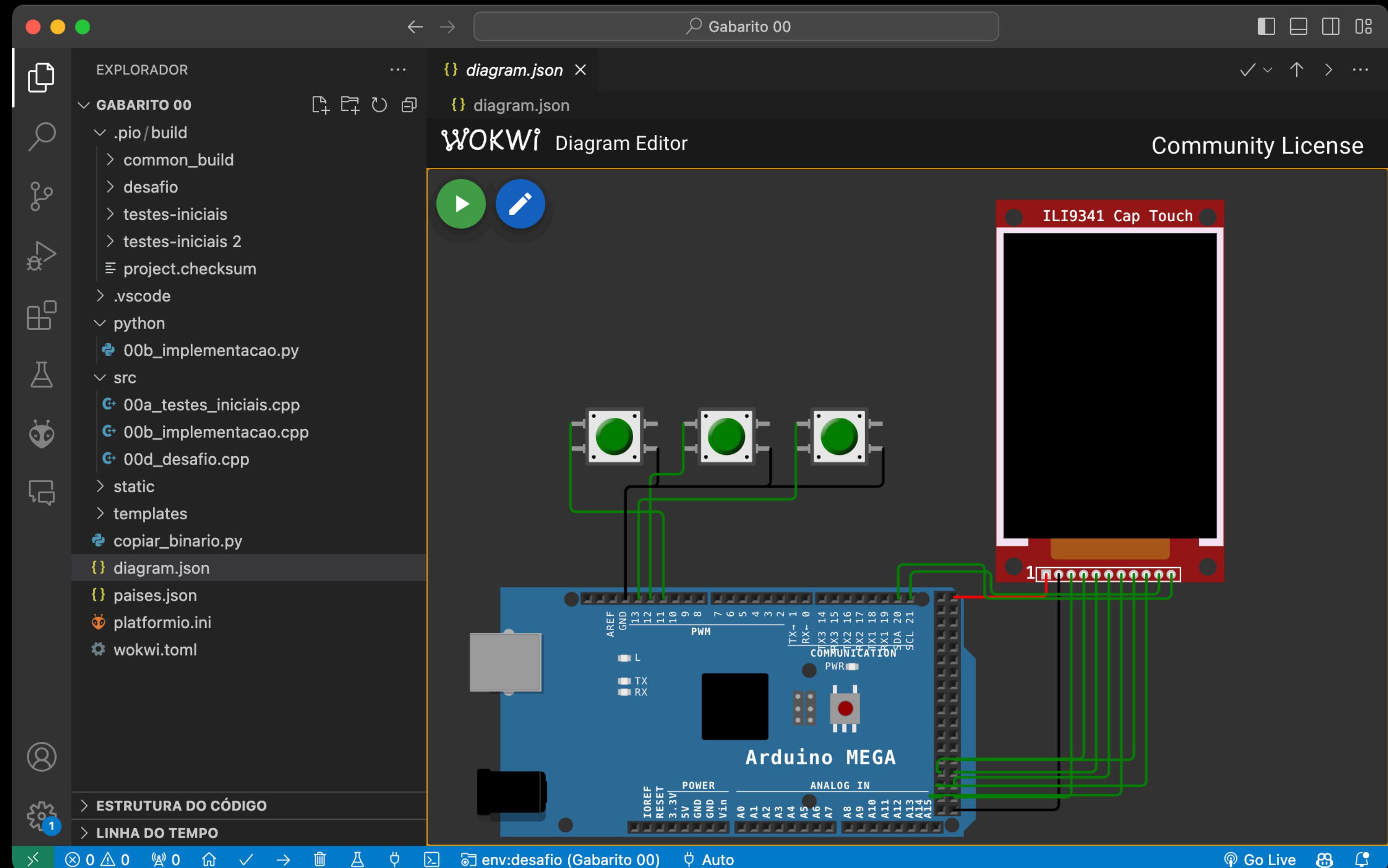
ENG4051 – Projeto Internet das Coisas

# Testes Iniciais



# Visual Studio Code

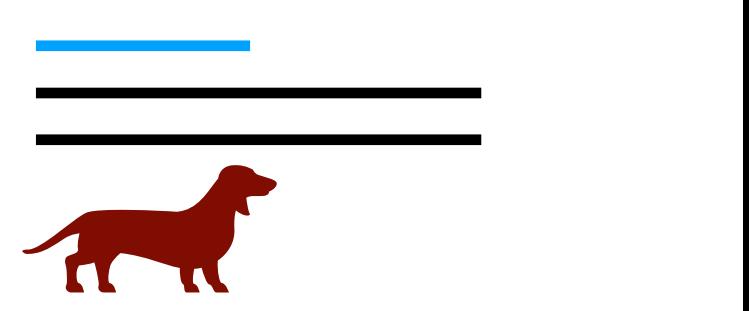
Programação no Visual Studio Code



Simulador Wokwi

Comece no código Python.

- 1 Crie uma página HTML (na pasta templates) com um link, dois parágrafos e uma imagem.  
Crie uma rota no Flask que renderize essa página.



- 2 Crie uma rota que receba uma string e envie uma mensagem para o Telegram com esse texto.

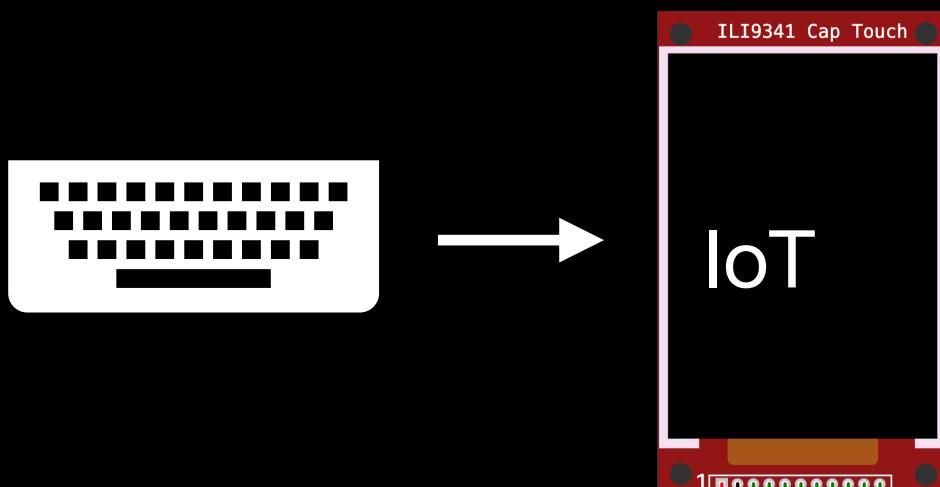


Vá para o código Arduino.

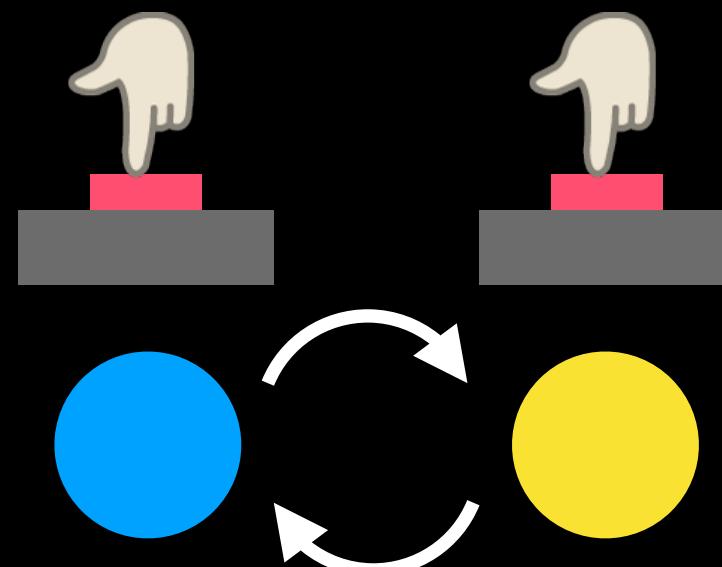
- 3 A cada 300 milissegundos, some 1 numa contagem e imprima na Serial se ela for menor ou igual a 20.  
↳ DICA: use a millis.

17  
18  
19  
20

Ao receber um texto da Serial, desenhe ele na tela.  
Apague o texto anterior antes de desenhar o novo.  
↳ DICA: desenhe um retângulo preto para apagar.



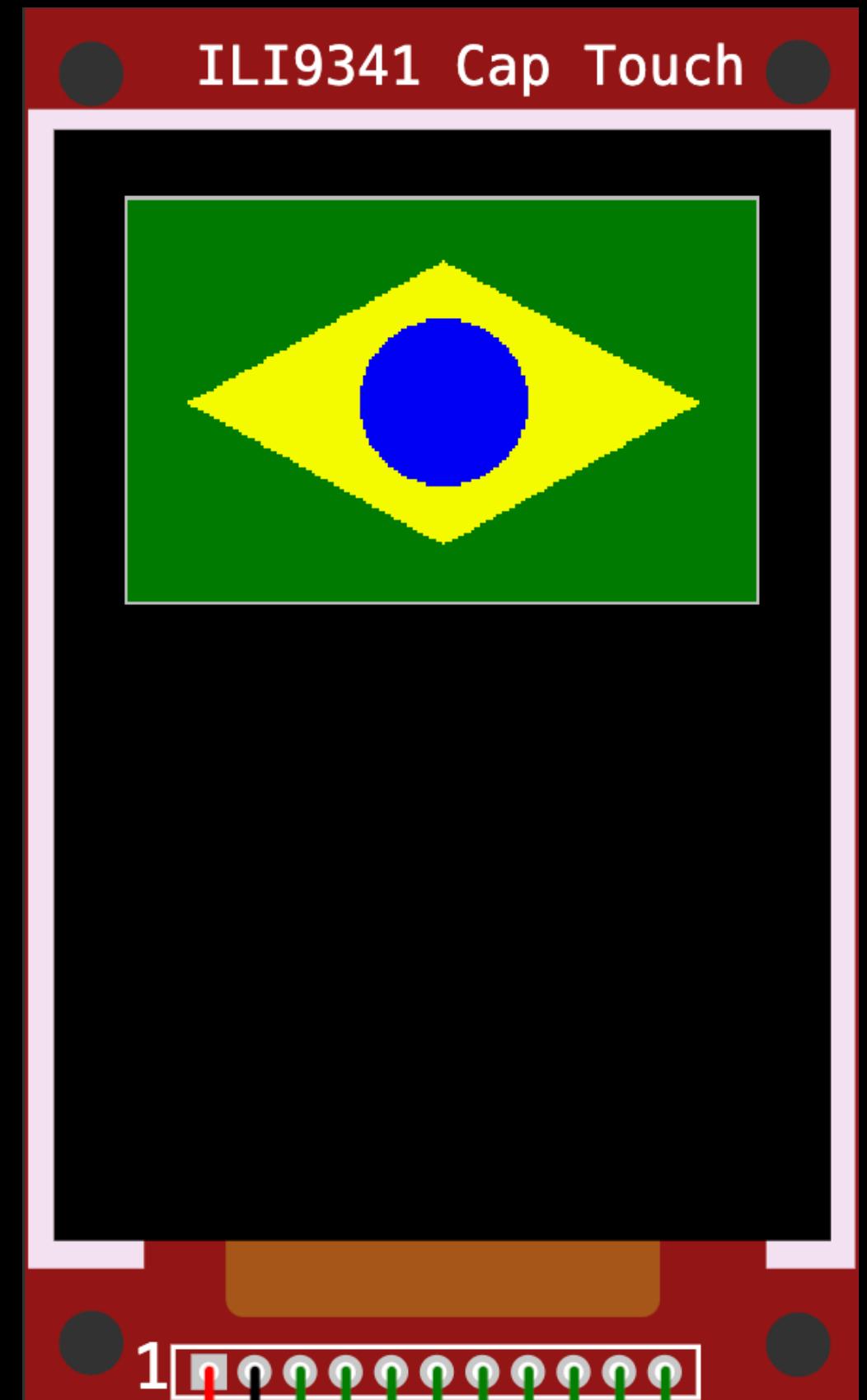
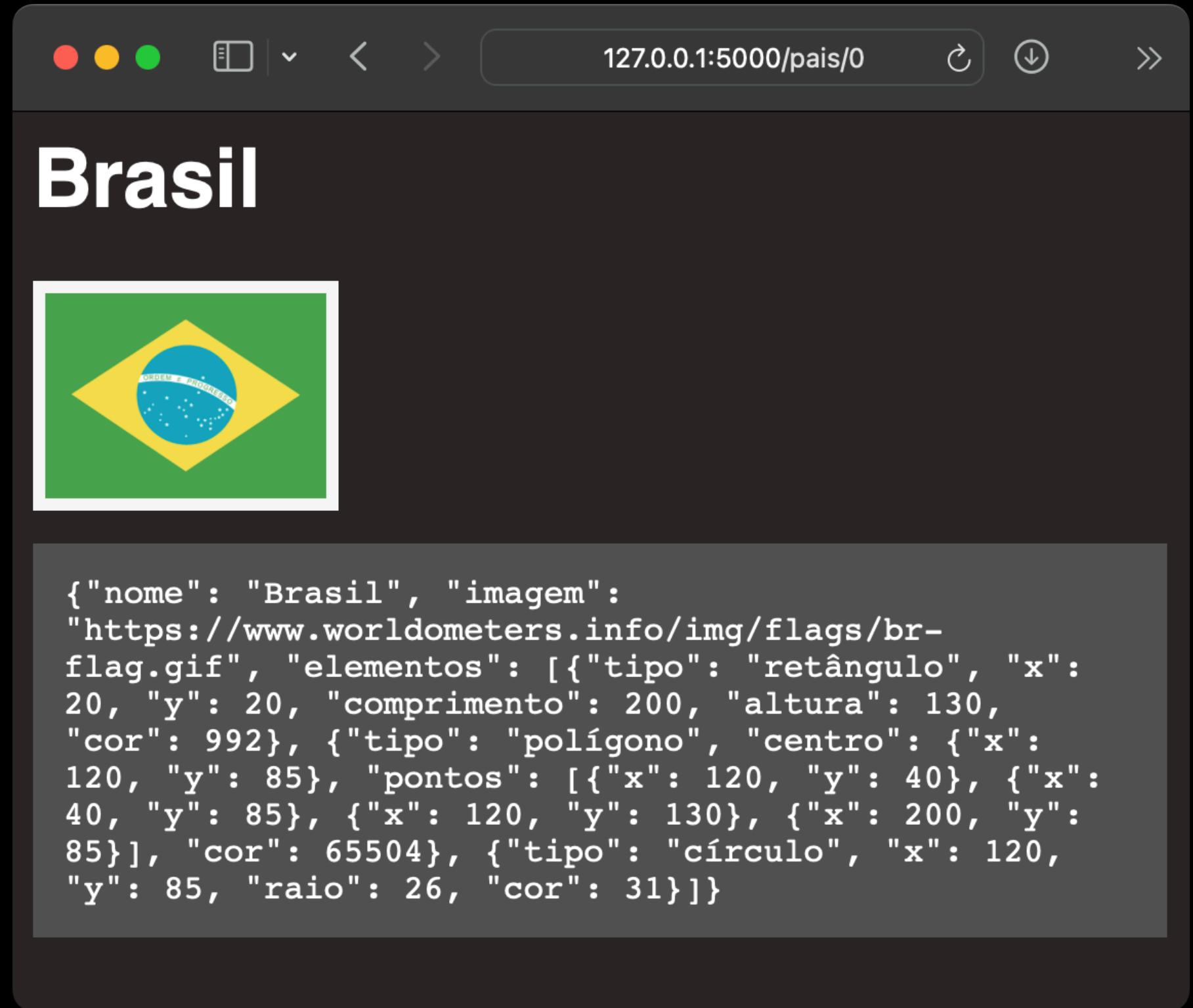
- 5 Ao apertar o botão 1 (pino 11), desenhe um círculo qualquer variando a cor entre azul e amarelo.  
↳ DICA: use uma variável global tipo bool.



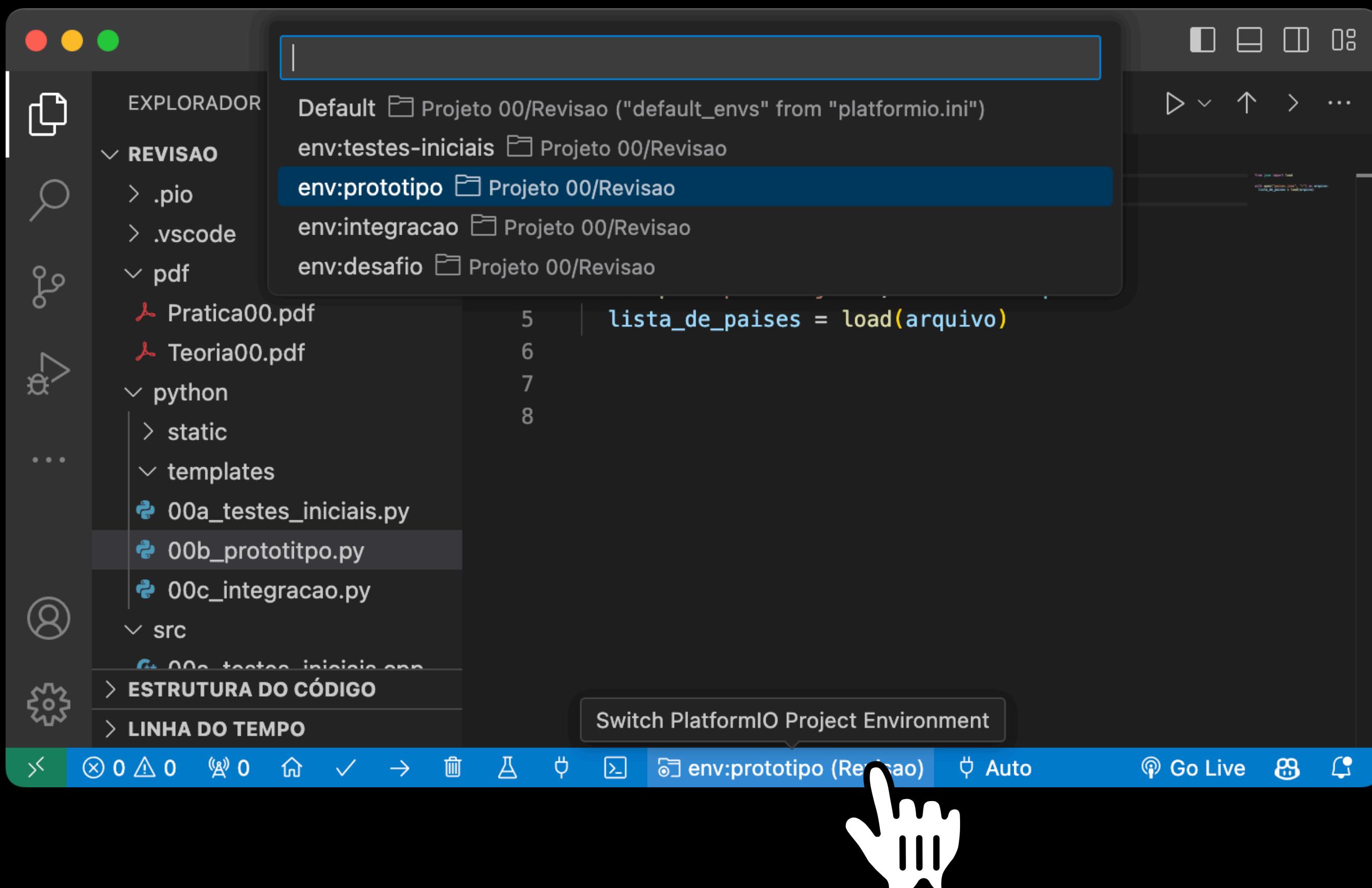
# Protótipo



Desenho de Bandeiras: Parte 2 – O Retorno



Página de Bandeiras + Desenho de Bandeiras na Tela



Mudança da Compilação para Próxima Etapa do Exercício

1

Crie uma rota que receba um índice, acesse os dados do país e mostre a página de detalhes com seu nome, bandeira e string JSON.

localhost:5000/pais/**0**

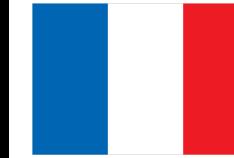
**Brasil**



```
{"nome": "Brasil", "imagem":  
"https://www.worldometers.info/  
img/flags/br-flag.gif", "elementos":  
{"tipo": "retângulo", "x": 20, "y":
```

localhost:5000/pais/**1**

**França**



```
{"nome": "França", "imagem":  
"https://www.worldometers.info/  
img/flags/fr-flag.gif", "elementos":  
{"tipo": "retângulo", "x": 20, "y":
```



2

Adicione o tag abaixo no HTML para mudar a aparência da página.

```
<link rel="stylesheet" href="/static/style.css">
```

**Brasil**



```
{"nome": "Brasil", "imagem":  
"https://www.worldometers.info/  
img/flags/br-flag.gif", "elementos":  
{"tipo": "retângulo", "x": 20, "y":
```



**Brasil**



```
{"nome": "Brasil", "imagem":  
"https://www.worldometers.info/img/  
flags/br-flag.gif", "elementos":  
{"tipo": "retângulo", "x": 20, "y":
```

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with the following interface elements:

- Top Bar:** Includes standard window controls (red, yellow, green buttons), a search bar labeled "Revisão", and icons for file operations like "New File", "Save", and "Close".
- Left Sidebar (Explorador):** Shows the project structure:
  - REVISÃO**: Contains ".pio", ".vscode", "python", and "src".
  - src**: Contains "00a\_testes\_iniciais.cpp", "00b\_implementacao.cpp" (which is selected and highlighted in blue), "00c\_aperfeicoamento.cpp", "00d\_desafio.cpp", ".gitignore", "copiar\_binario.py", "diagram.json", "países.json", "platformio.ini", and "wokwi.toml".
- Central Editor Area:** Displays the content of "00b\_implementacao.cpp". The code is written in C++ and defines functions to draw rectangles, circles, and polygons based on JSON documents. A large rectangular selection box highlights the entire function definitions for "desenharRetangulo", "desenharCirculo", and "desenharPoligono".
- Bottom Status Bar:** Shows file status indicators (0 changes, 0 warnings, 0 errors), file name ("Default (Revisão)", "Auto"), line and column information ("Ln 3, Col 30"), encoding ("UTF-8"), line endings ("CRLF"), code completion ("{}"), C++ mode ("C++"), "Go Live" button, PlatformIO icon, and a bell icon.

Código Inicial do Arduino

3

Implemente a função de desenhar um retângulo

desenharRetangulo (JsonDocument elemento)

```
{  
    "x": 10,  
    "y": 30,  
    "altura": 60,  
    "comprimento": 100,  
    "cor": 992  
}
```

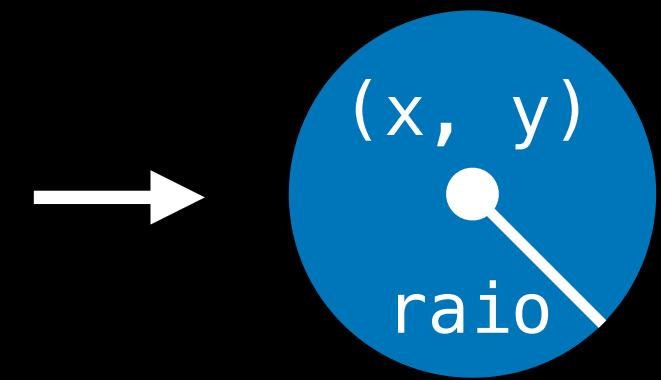


4

Implemente a função de desenhar um círculo

desenharCirculo (JsonDocument elemento)

```
{  
    "x": 70,  
    "y": 28,  
    "raio": 45,  
    "cor": 31  
}
```

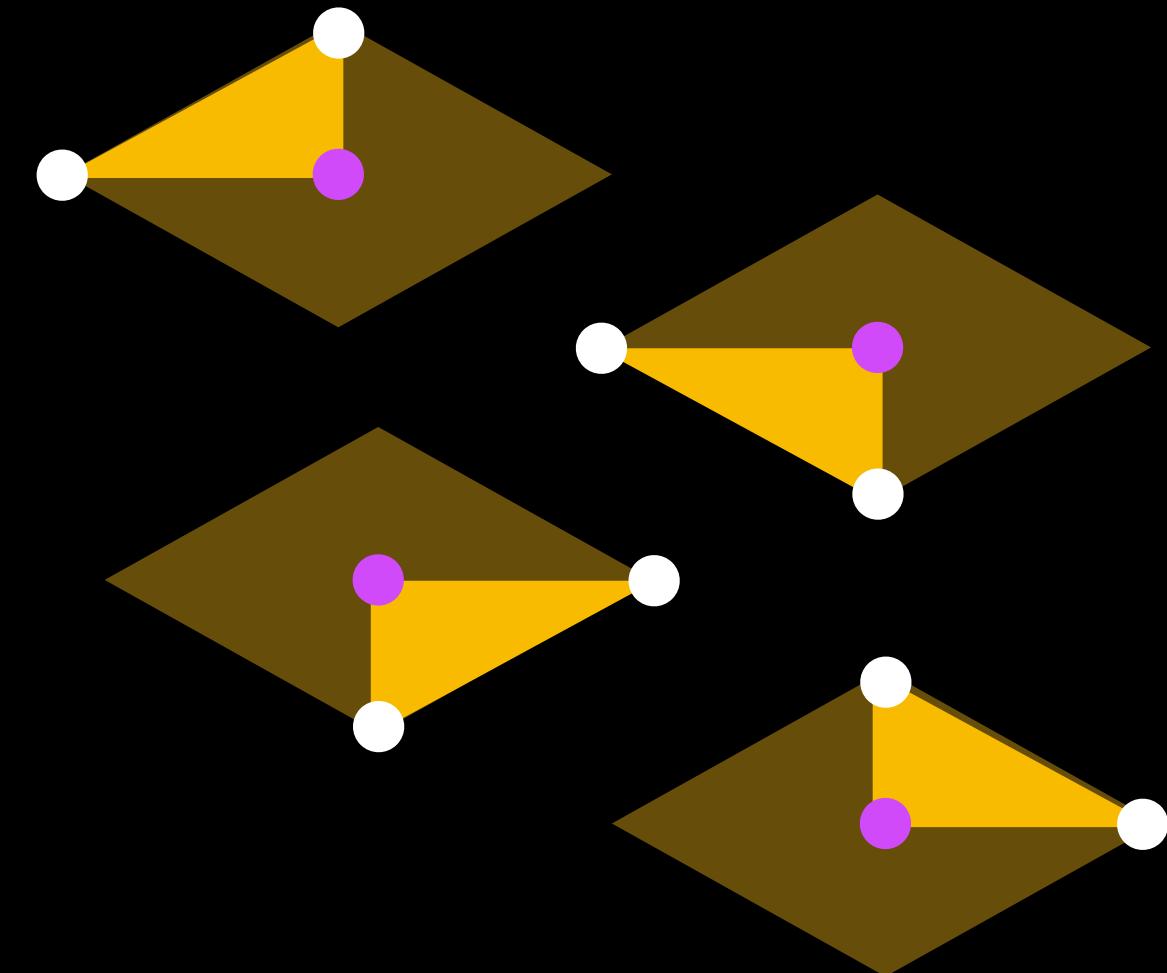
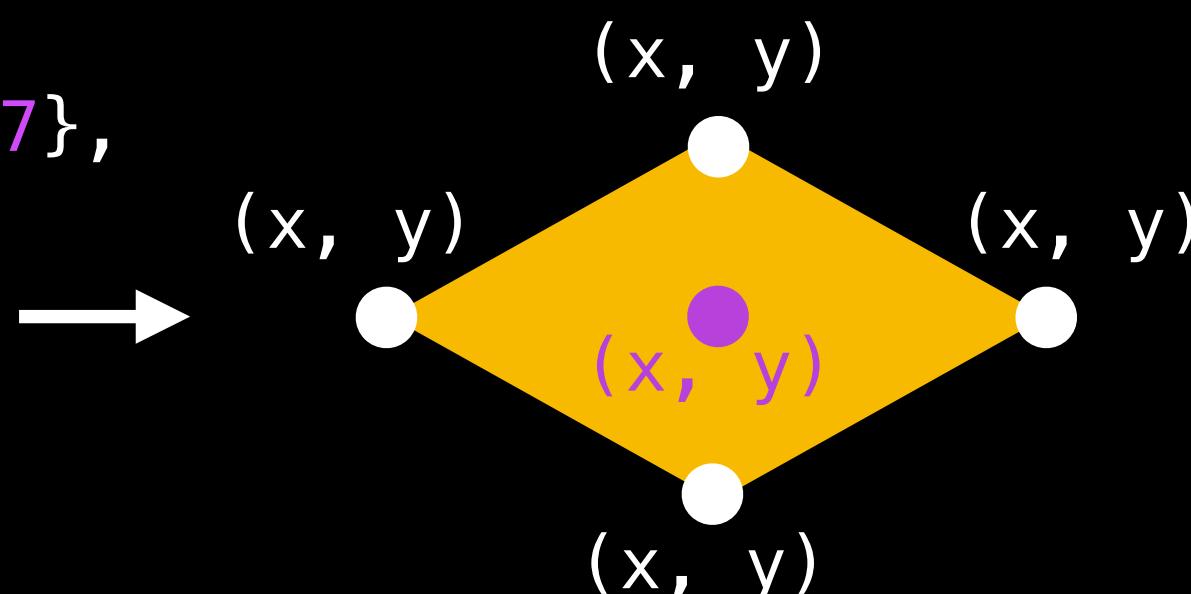


5

Implemente a função de desenhar um polígono.  
Pinte triângulos com o ponto central, o ponto atual e o anterior

desenharPoligono (JsonDocument elemento)

```
{  
    "centro": {"x": 5, "y": 7},  
    "pontos": [  
        {"x": 5, "y": 0},  
        {"x": 0, "y": 7},  
        ...  
    ],  
    "cor": 992  
}
```



Funções de Desenho

# Integração

1

Crie uma página com um formulário para adicionar uma nova bandeira

Nome da Bandeira
URL da Imagem
JSON dos Elementos

**Enviar**

2

Ao receber os dados do formulário, adicione um novo dicionário na lista de países.

↪ DICA: use a `json.loads` para converter o texto {  
do terceiro campo para uma lista de elementos.

Japao

<https://www.worldometers.info/img/flags/ja-flag.gif>

[{"tipo": "retângulo", "x": 20, "y": 20, "comprimento": 200, "altura": 130, "cor": 65535},  
 {"tipo": "círculo", "x": 120, "y": 85, "raio": 40, "cor": 63488}]

"nome": "Japao",  
"imagem": "https://www.w...","  
"elementos": [{...}, {...}]

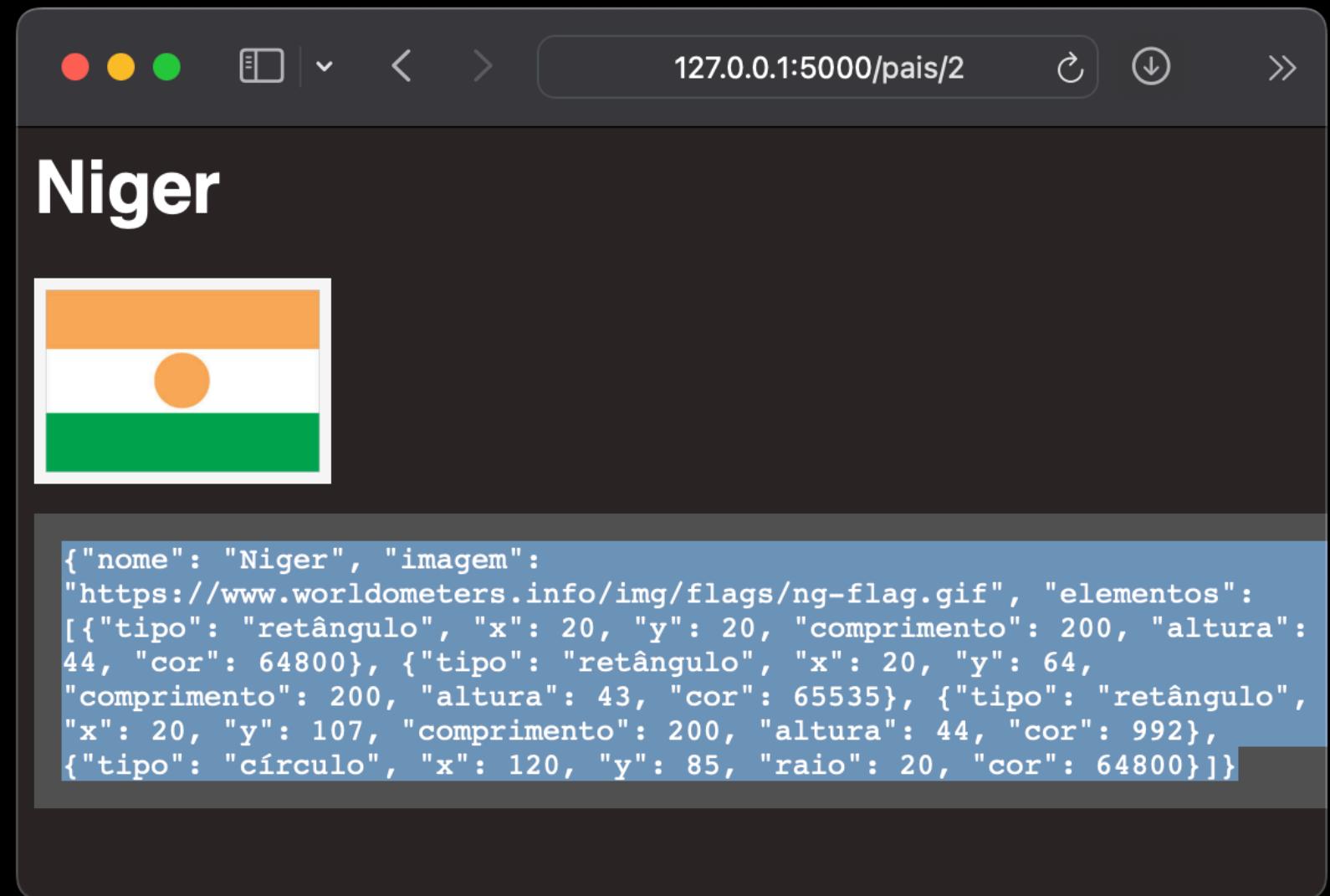
3

Também ao receber os dados, baixe a imagem da bandeira e envie para o Telegram

'[https://www.worldometers.info/  
img/flags/ac-flag.gif](https://www.worldometers.info/img/flags/ac-flag.gif)'



Adição de Bandeiras no Servidor



127.0.0.1:5000/pais/2

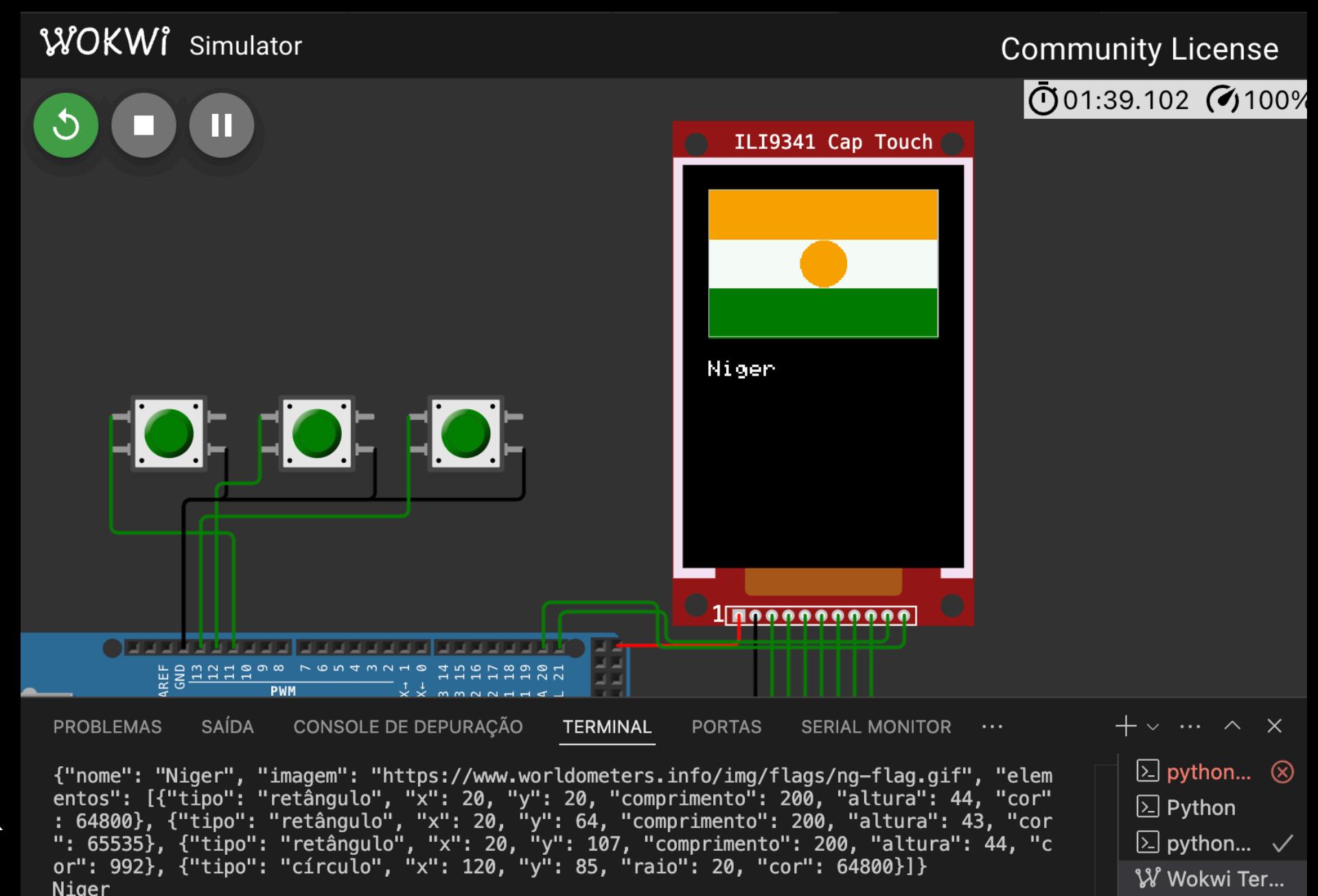
# Niger



```
{"nome": "Niger", "imagem": "https://www.worldometers.info/img/flags/ng-flag.gif", "elementos": [{"tipo": "retângulo", "x": 20, "y": 20, "comprimento": 200, "altura": 44, "cor": 64800}, {"tipo": "retângulo", "x": 20, "y": 64, "comprimento": 200, "altura": 43, "cor": 65535}, {"tipo": "retângulo", "x": 20, "y": 107, "comprimento": 200, "altura": 44, "cor": 992}, {"tipo": "círculo", "x": 120, "y": 85, "raio": 20, "cor": 64800}]}{}
```

Ctrl+C

Ctrl+V

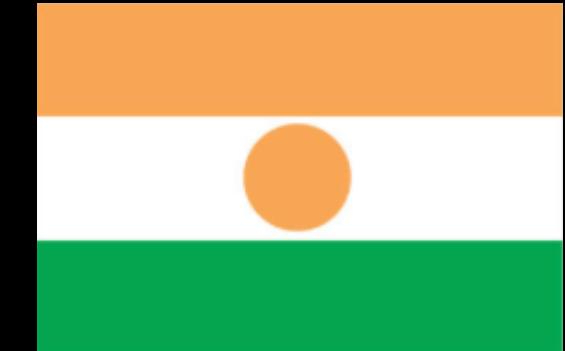


Cópia do JSON para o Arduino via Serial

4

Ao receber um texto da Serial, converta para JsonDocument, percorra a lista de elementos e chame as funções da Implementação

```
{"nome": "Niger", "imagem": "https://www.worldometers.info/img/flags/ng-flag.gif",  
"elementos": [{"tipo": "retângulo", "x": 20, "y": 20, "comprimento": 200, "altura": 44, "cor": 64800},  
 {"tipo": "retângulo", "x": 20, "y": 64, "comprimento": 200, "altura": 43, "cor": 65535}, {"tipo": "retângulo", "x": 20, "y": 107, "comprimento": 200, "altura": 44, "cor": 992},  
 {"tipo": "círculo", "x": 120, "y": 85, "raio": 20, "cor": 64800}]}
```



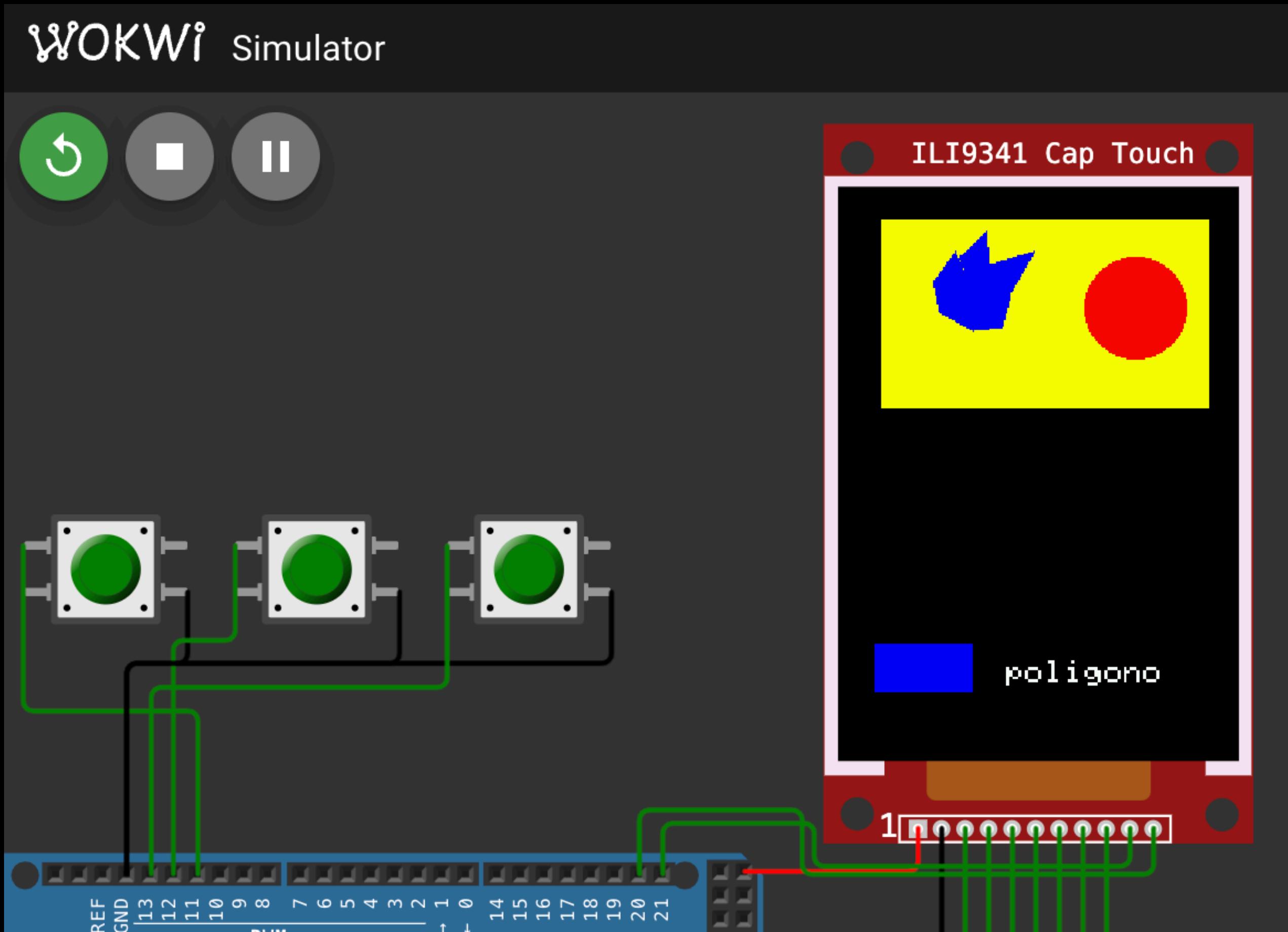
5

Escreva o nome do país abaixo da bandeira, apagando o nome da bandeira anterior.



Niger

# Desafio Extra



Desenho Manual de Uma Bandeira

```
unsigned long instanteAnteriorToque;

void loop() {
  if (touch.touched()){
    TS_Point p = touch.getPoint();

    // ajusta coordenadas para casar com as do desenho
    p.x = map(p.x, 0, 240, 240, 0);
    p.y = map(p.y, 0, 320, 320, 0);

    // debounce para não registrar vários toques
    if (millis() > instanteAnteriorToque + 300)
    {
      instanteAnteriorToque = millis();

      // Seu código para tratar o toque...
    }
  }
}
```

Exemplo de Código para Capturar Toque na Tela

1

Ao clicar no botão 1, alterne umas 6 cores.  
Desenhe um retângulo com a cor atual no fundo da tela.



2

Ao clicar no botão 2, alterne entre "retangulo", "circulo" e "poligono".  
Escreva o tipo atual no fundo da tela.

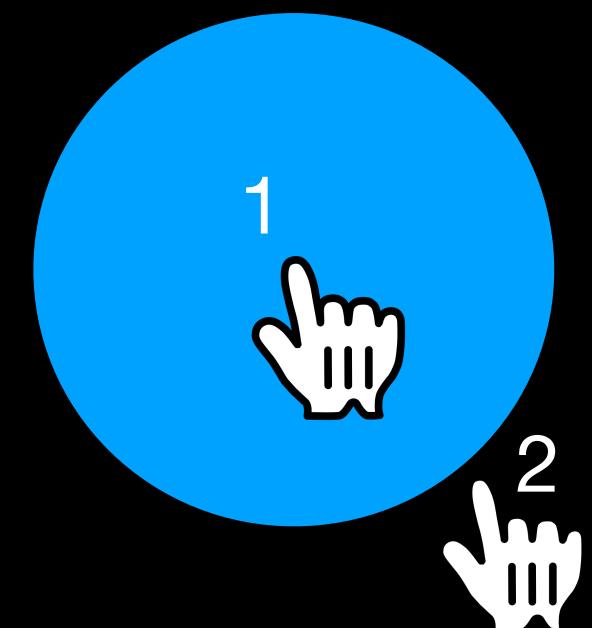
3

Se o modo for "retangulo", toque em dois pontos  
para desenhar um com a cor atual.



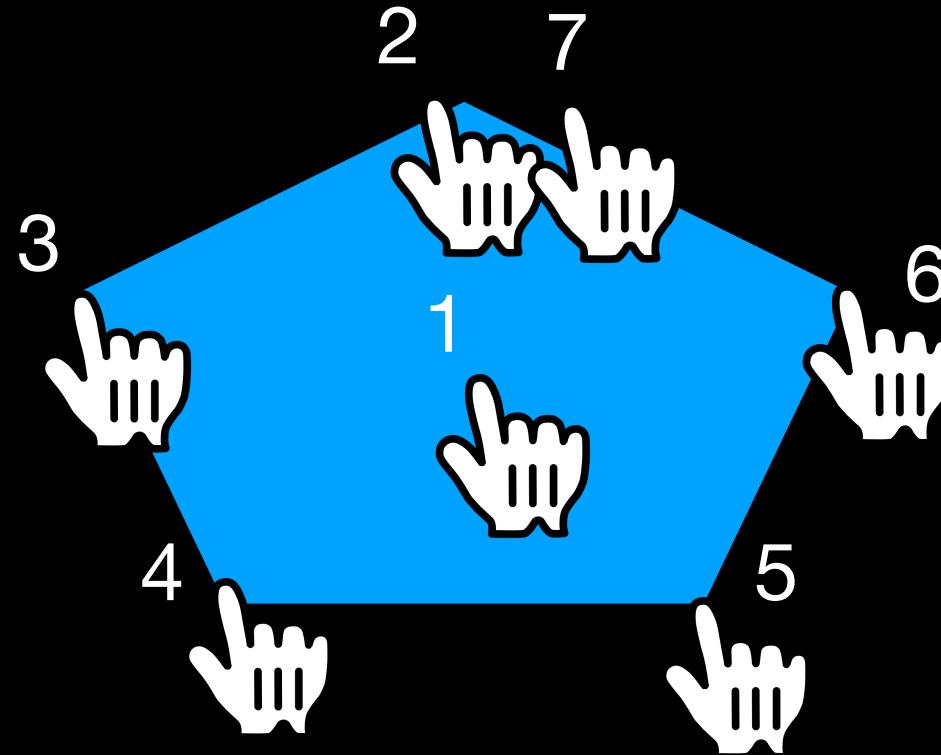
4

Se o modo for "circulo", toque em dois pontos  
para desenhar um com a cor atual.  
DICA: use a sqrt e a pow para calcular o raio.



**5**

Se o modo for "polígono", toque uma vez para guardar o ponto central.  
Depois, ao ir tocando nos N vértices, desenhe triângulos.  
Ao tocar próximo do primeiro ponto, finalize a forma.



**6**

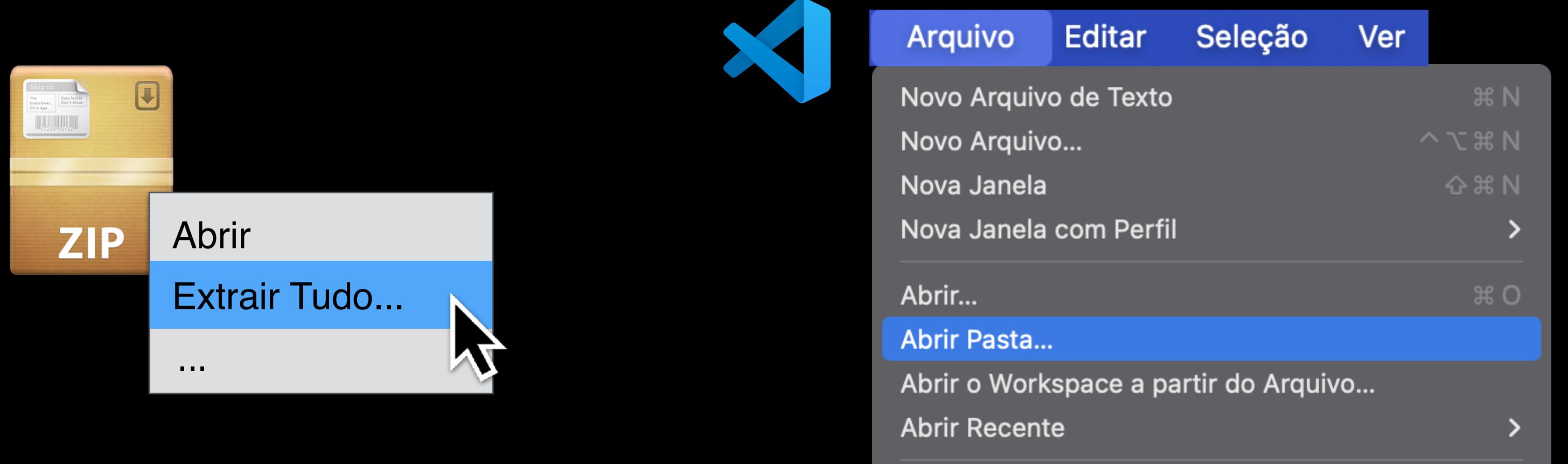
Desenha a bandeira da Mauritânia



Desenho de Polígono + Nova Bandeira



Material da Introdução

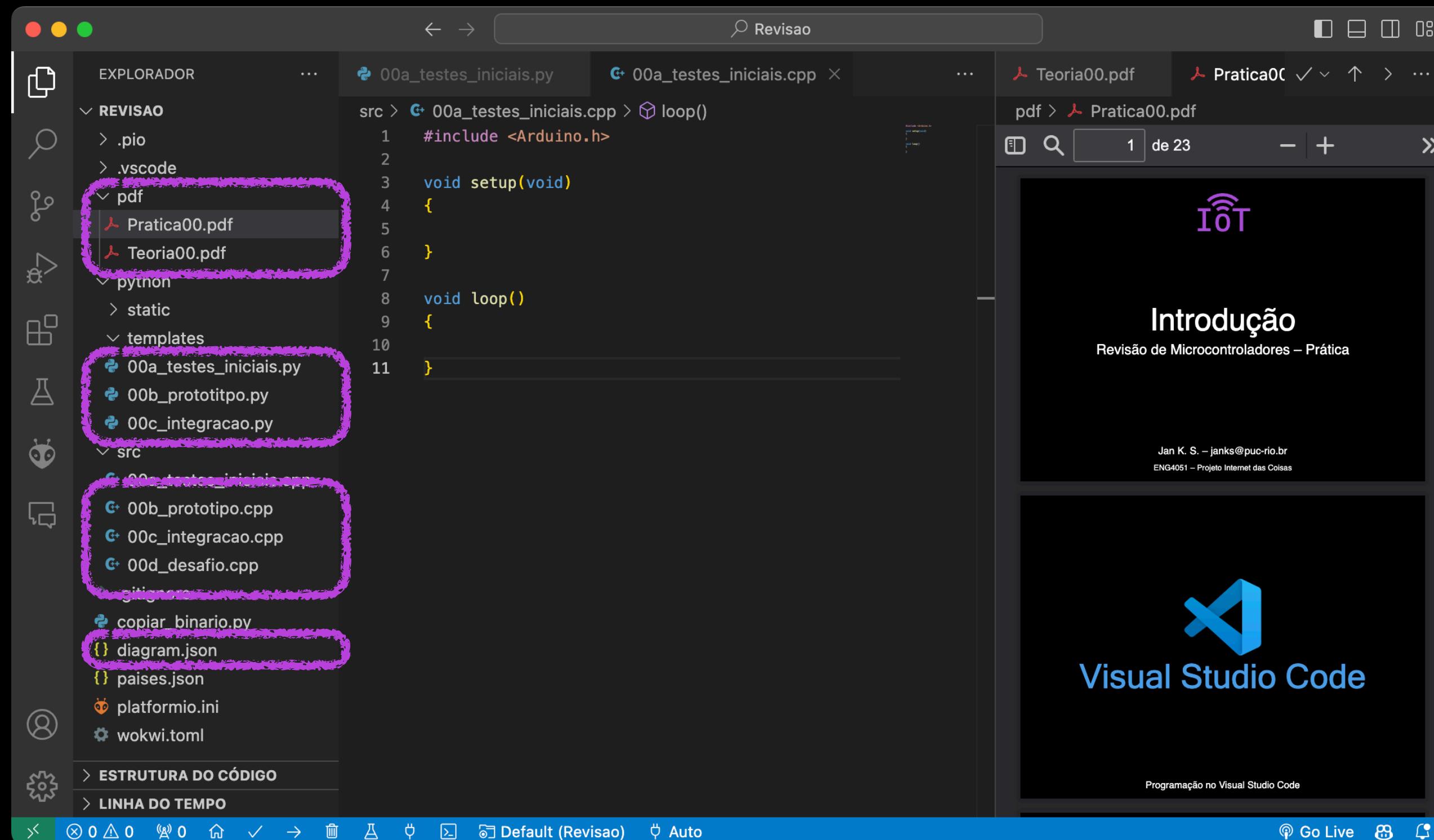


PDFs

Python

Arduino

Simulador



Extração de Arquivos + Visualização no VS Code



[janks.link/iot/introducao.zip](https://janks.link/iot/introducao.zip)

Material da Introdução