



Projeto 01

Coisas Configuráveis – Prática

Jan K. S. – janks@puc-rio.br

ENG4051 – Projeto Internet das Coisas

Testes Iniciais

1

A cada 1 segundo (usando a millis no loop), obtenha a data/hora atual e imprima o horário no formato ao lado.
↳ DICA: veja a senha do WiFi no arquivo credenciais.txt.

Agora são 16:30:01
Agora são 16:30:02
Agora são 16:30:03
Agora são 16:30:04

2

Crie uma rota que receba 3 parâmetros entre 0 e 255. Use esses valores para mudar a cor do LED RGB.
↳ DICA: converta os parâmetros para int e lembre-se de enviar uma resposta 200 de volta ao navegador.

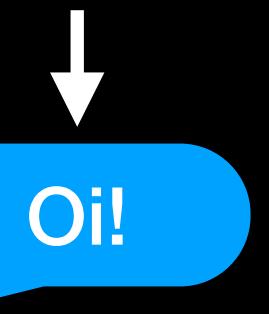
192.168.X.X/cor/**50/0/0**



3

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

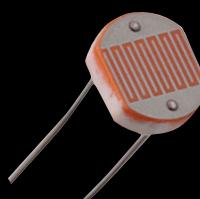
192.168.X.X/msg/**Oi!**



4

Crie um arquivo HTML na pasta data com um parágrafo contendo um parâmetro {{valorDaLuz}}.
Construa o sistema de arquivos e suba para o ESP32.

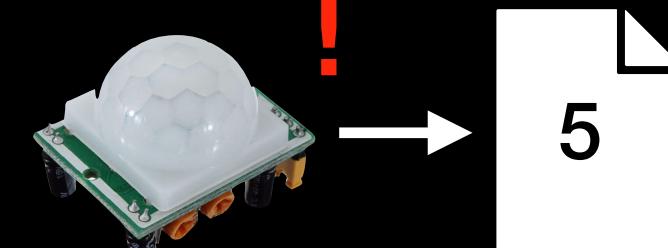
192.168.X.X/luz



Luz: 73%

5

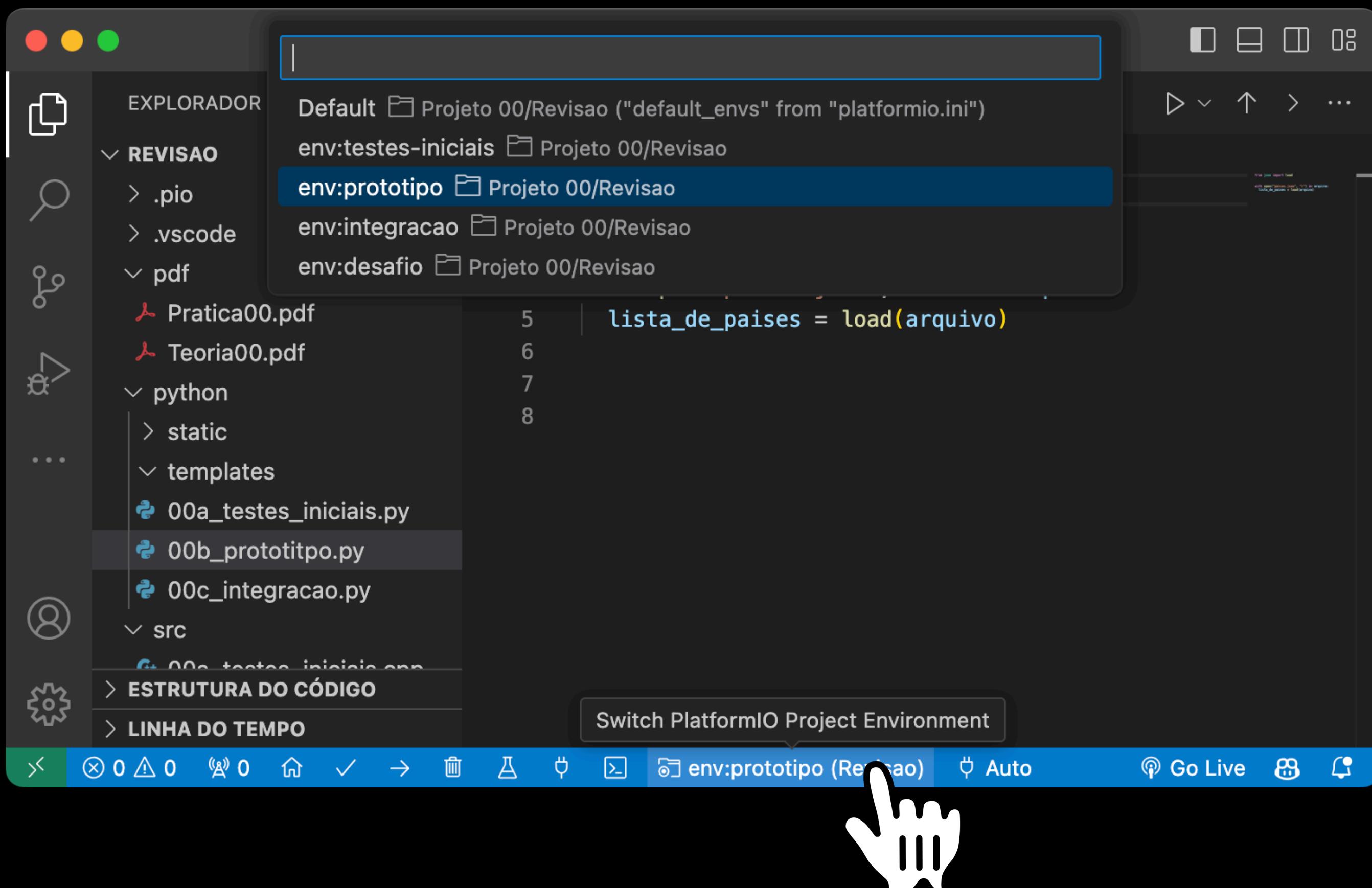
Ao detectar movimento, leia o número no contagem.txt, adicione 1 nele, imprima o valor resultante na Serial e escreva de volta no arquivo.



Protótipo



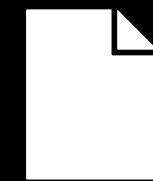
```
{  
  "corRGB": "vermelho",  
  "alertarMovimento": true,  
  "horarioAcenderTudo": "17:30:00",  
  "horarioApagarTudo": "17:30:10",  
}
```



Mudança da Compilação para Próxima Etapa do Exercício

1

Na setup, carregue o arquivo `ajustes.json` numa variável `JsonDocument` global.



→ `JsonDocument ajustes;`
`ajustes.json`

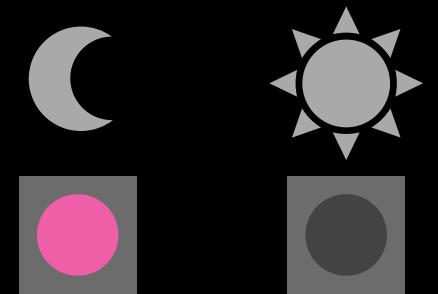
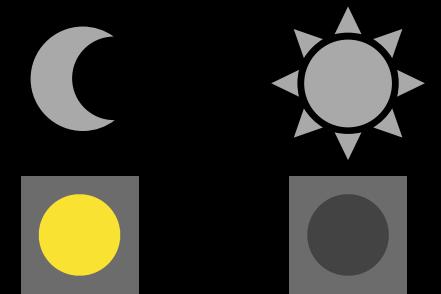
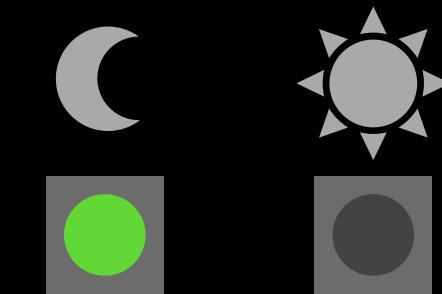
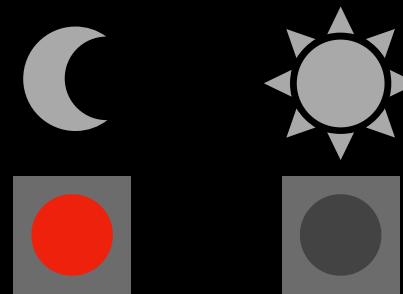
2

Monitore o sensor de luz na loop.

Se estiver escuro, acenda o LED RGB com a cor configurada nos ajustes.

Se estiver claro, apague o LED RGB.

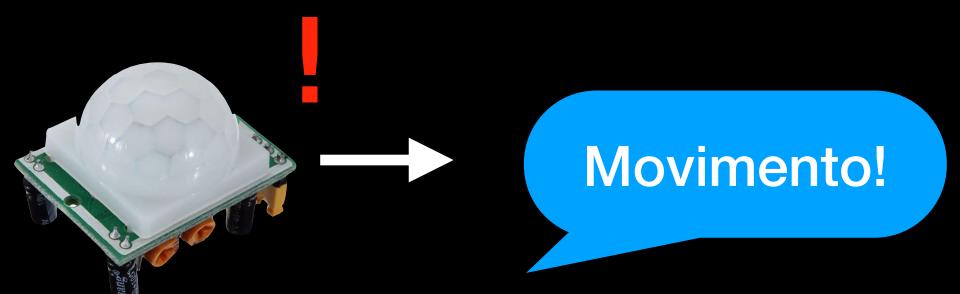
```
{ "corRGB": "vermelho", "corRGB": "verde", "corRGB": "amarelo", "corRGB": "rosa",  
... }
```



3

Ao detectar movimento,
envie um alerta para o Telegram
se isso estiver configurado como `true` nos ajustes.

```
{ "alertarMovimento": true, ... }
```



4

Monitore o horário atual na loop.

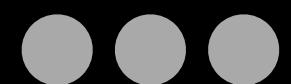
Acenda ou apague todos os LEDs quando o horário atual coincidir com os horários do ajuste.

```
{  
  "horarioAcenderTudo": "17:30:00",  
  "horarioApagarTudo": "17:30:10",  
  ...  
}
```

17:30:00



17:30:10



Protótipo

Integração

Configurações

LEDs

Controle

| | | |
|-------|-------------------------|------------------------|
| LED 1 | Acender | Apagar |
| LED 2 | Acender | Apagar |
| LED 3 | Acender | Apagar |

Hora de Acender Hora de Apagar
17:30:00 17:30:10

LED RGB

- Vermelho
- Verde
- Amarelo
- Rosa

Sensor de Movimento

Alertar movimento no Telegram

Salvar

Configuração da Automação pelo Navegador

1

Crie uma rota que renderize o arquivo `ajustes.html`, atualizando os parâmetros de acordo com os ajustes.

{

```
{"horarioAcenderTudo": "17:30:00",  
 "horarioApagarTudo": "17:30:10",  
 "corRGB": "vermelho",  
 "alertarMovimento": true  
}
```

```
"{{horarioAcenderTudo}}" → "17:30:00"  
"{{horarioApagarTudo}}" → "17:30:10"  
"{{opcaoVermelho}}" → "checked"  
"{{opcaoAlerta}}" → "checked"
```

Vermelho
 Verde
 Amarelo
 Rosa

Hora de Acender
17:30:00
Hora de Apagar
17:30:10

Alertar movimento no Telegram

Crie a versão HTTP_POST da rota da página de ajustes.

2

Ao receber os dados do formulário, atualize os ajustes da variável global e o arquivo de configuração.

Em seguida, redirecione de volta para a própria página de ajustes.

Salvar



| | |
|--------------------|----------|
| horarioAcenderTudo | 17:45:00 |
| horarioApagarTudo | 17:45:15 |
| corRGB | amarelo |
| alertarMovimento | true |



ajustes.json

```
JsonDocument ajustes;
```

3

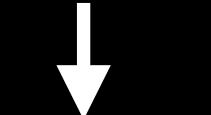
Crie uma rota que receba um texto e um índice.

Se o texto for "ligar", acenda o LED daquele índice.

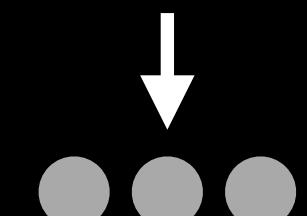
Se o texto for "desligar", apague o LED.

Em seguida, redirecione de volta para a página de ajustes.

192.168.X.X/led/**ligar**/0



192.168.X.X/led/**desligar**/0



Integração

Desafio Extra

| Data | LED 1 | LED 2 | LED 3 |
|---------------------|---------|---------|---------|
| 17/03/2025 15:30:11 | aceso | apagado | aceso |
| 17/03/2025 15:33:42 | apagado | aceso | aceso |
| 17/03/2025 15:37:21 | apagado | aceso | apagado |



Histórico do Estado dos LEDs no Navegador e no Telegram

1

Crie um JsonDocument global para a lista de histórico do LEDs. Ao acender ou apagar um LED, guarde os estados e a data/hora num dicionário e adicione-o na lista.

Caso a lista tenha mais de 10 itens, remova o mais antigo.

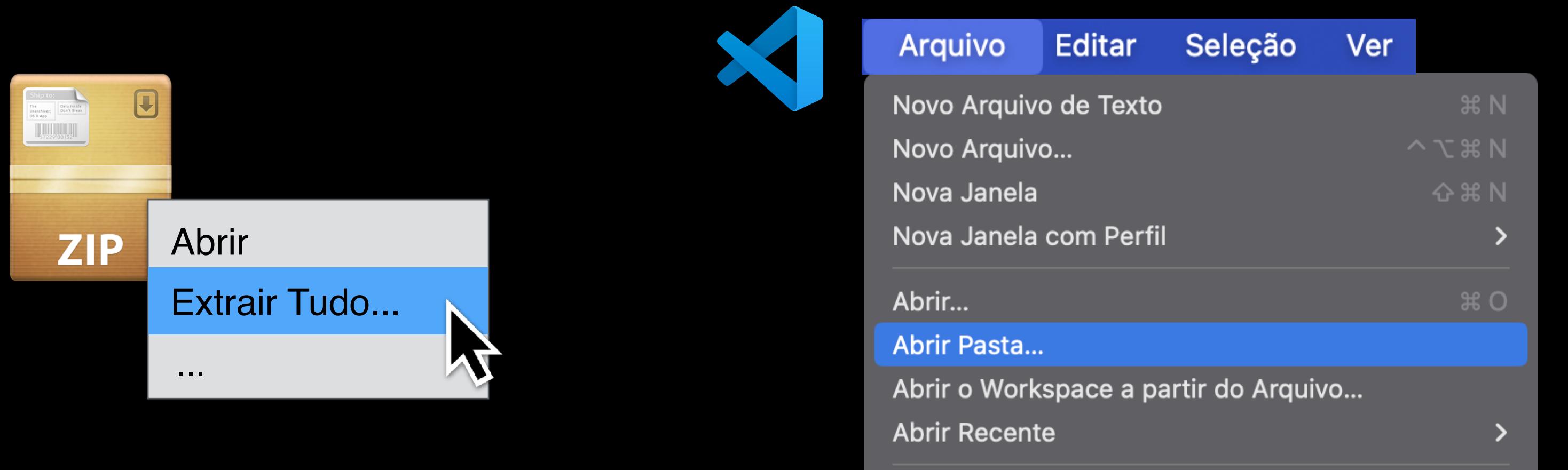
```
[  
    {"data": "17/03 15:30:11", "led1": "X", "led2": "V", "led3": "V"},  
    {"data": "17/03 15:33:42", "led1": "V", "led2": "X", "led3": "V"},  
    ...  
]
```

2

A cada 15 segundos, envie um texto para o Telegram com o histórico das luzes no formato HTML abaixo.

↪ DICA: use a propriedade parse_mode como html no JSON do Telegram.

```
<pre>  
|       Data       | 1 | 2 | 3 |  
-----  
| 17/03 15:30:11 | X | V | V |  
| 17/03 15:33:42 | V | X | V |  
...  
</pre>
```



Arquivos
PDFs
Código

```
Projeto01
├── data
│   ├── ajustes.html
│   ├── ajustes.json
│   └── contagem.txt
└── pdfs
    ├── Pratica01.pdf
    ├── Resumo01.pdf
    └── Teoria01.pdf
src
    ├── 01a_testes_iniciais.cpp
    ├── 01b_prototipo.cpp
    ├── 01c_integracao.cpp
    └── 01d_desafio.cpp
```

01a_testes_iniciais.cpp

```
#include <Arduino.h>

void setup() {
    Serial.begin(115200); delay(500);
    Serial.println("\nProjeto 01 - Testes Iniciais");
}

void loop() {
```

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure with folders "PROJETO01", "data", "pdfs", and "src". The "src" folder contains four files: "01a_testes_iniciais.cpp", "01b_prototipo.cpp", "01c_integracao.cpp", and "01d_desafio.cpp". The "01a_testes_iniciais.cpp" file is currently selected and shown in the main editor area. The code editor displays the following C++ code:

Extração de Arquivos + Visualização no VS Code



janks.link/iot/projeto01.zip

Material do Projeto 01