

Monitoramento de Enfestos com ESP32 e Análise de Dados com Python e SQLite

Descrição

Este projeto tem como objetivo simular um processo de **monitoramento de enfesto industrial**, utilizando um **sensor ultrassônico conectado ao ESP32**, cuja leitura é capturada via protocolo RFC2217 e registrada em um banco de dados local (SQLite3).

Os dados capturados incluem distância, data/hora, código da máquina e ordem de produção. A partir disso, o sistema identifica o número de “folhas” movimentadas no enfesto (ida e volta da máquina) e permite:

- Exportação dos dados em CSV e JSON
- Visualizações gráficas interativas com Plotly
- Análises de produtividade por máquina e OP
- Registro automatizado em banco de dados com persistência

Integrantes do Grupo

Thiago Scutari – RM562831

Henrique Ribeiro Siqueira – RM565044

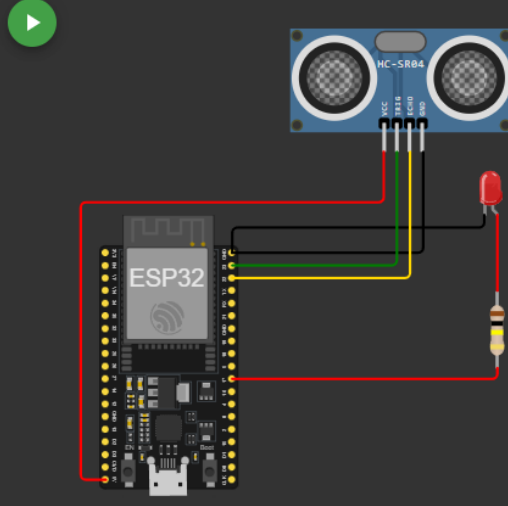
Mariana Cavalcante Oliveira – RM561678

Repositório do Projeto

 github.com/ThiagoScutari/sprint_2

Dados capturados do Sensor pelo código python

WOKWI Simulator



PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
2025-06-13 23:14:44,242.9
2025-06-13 23:14:45,113.1
2025-06-13 23:14:45,31.0
2025-06-13 23:14:46,3.0
2025-06-13 23:14:46,175.9
2025-06-13 23:14:47,299.0
2025-06-13 23:14:47,390.9
2025-06-13 23:14:48,317.0
2025-06-13 23:14:48,201.0
2025-06-13 23:14:49,66.9
2025-06-13 23:14:49,2.0
2025-06-13 23:14:50,105.0
2025-06-13 23:14:50,271.0
2025-06-13 23:14:51,344.9
2025-06-13 23:14:51,399.9
2025-06-13 23:14:52,236.0
2025-06-13 23:14:52,87.9
2025-06-13 23:14:53,21.0
2025-06-13 23:14:53,42.0
2025-06-13 23:14:54,183.0
2025-06-13 23:14:55,323.9
2025-06-13 23:14:55,399.9
2025-06-13 23:14:56,399.9
```

OUTLINE

python > src > monitorar_sensor.py > ...

```
38
39 def processar_linha(linha, estado, db):
40     try:
41         if ',' in linha:
42             dataHora_str, distancia_str = linha.split(',')
43             dataHora = datetime.strptime(dataHora_str.strip(), "%Y-%m-%d %H:%M:%S")
44             distancia = float(distancia_str.strip())
45
46             folhas = detectar_folha(distancia, estado)
47
48             logging.info(f"[{dataHora}] {distancia:.1f} cm | OP={ORDEM_PRODUCAO}")
49
50             db.inserir_leitura(
51                 codMaquina=COD_MAQUINA,
52                 ordemProducao=ORDEM_PRODUCAO,
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
Leitura inserida com sucesso: maq002 | OP=OP00221 | 399.9cm | folhas=0
2025-06-13 23:26:56,204 [INFO] [2025-06-13 23:27:01] 400.0 cm | OP=OP00221 | folhas=0
Leitura inserida com sucesso: maq002 | OP=OP00221 | 400.0cm | folhas=0
2025-06-13 23:26:56,716 [INFO] [2025-06-13 23:27:01] 399.9 cm | OP=OP00221 | folhas=0
Leitura inserida com sucesso: maq002 | OP=OP00221 | 399.9cm | folhas=0
2025-06-13 23:26:57,244 [INFO] [2025-06-13 23:27:02] 399.9 cm | OP=OP00221 | folhas=0
Leitura inserida com sucesso: maq002 | OP=OP00221 | 399.9cm | folhas=0
2025-06-13 23:26:57,766 [INFO] [2025-06-13 23:27:02] 399.9 cm | OP=OP00221 | folhas=0
Leitura inserida com sucesso: maq002 | OP=OP00221 | 399.9cm | folhas=0
2025-06-13 23:26:58,290 [INFO] [2025-06-13 23:27:03] 400.0 cm | OP=OP00221 | folhas=0
Leitura inserida com sucesso: maq002 | OP=OP00221 | 400.0cm | folhas=0
2025-06-13 23:26:58,817 [INFO] [2025-06-13 23:27:03] 399.9 cm | OP=OP00221 | folhas=0
Leitura inserida com sucesso: maq002 | OP=OP00221 | 399.9cm | folhas=0
2025-06-13 23:26:59,340 [INFO] [2025-06-13 23:27:04] 399.9 cm | OP=OP00221 | folhas=0
Leitura inserida com sucesso: maq002 | OP=OP00221 | 399.9cm | folhas=0
```

Gravação em banco de dados SQLite3

EXPLORER

SPRINT_2

docs

ESP32_Firmware

python

assets

database

enfesto.db

src

__pycache__

output

leituras.csv

leituras.json

analise_dados.py

db_manager.py

main.py

monitorar_sensor.py

venv

Include

db_manager.py

analise_dados.py

main.py

README.md

monitorar_sensor.py

enfesto.db

python > database > enfesto.db

Filter

Rows: 477

Filter 477 rows...

Upgrade to PRO

TABLES

leituras

sqlite_seque...

	id	codMaqu...	ordemPr...	dataHora	distancia	folhas
	1	1 maq001	OP00123	2025-06-13 21:52:44	2	0
	2	2 maq001	OP00123	2025-06-13 21:52:45	2	0
	3	3 maq001	OP00123	2025-06-13 21:52:45	2	0
	4	4 maq001	OP00123	2025-06-13 21:52:46	2	0
	5	5 maq001	OP00123	2025-06-13 21:52:46	2	0
	6	6 maq001	OP00123	2025-06-13 21:52:47	2	0
	7	7 maq001	OP00123	2025-06-13 21:52:47	2	0
	8	8 maq001	OP00123	2025-06-13 21:52:48	2	0
	9	9 maq001	OP00123	2025-06-13 21:52:48	2	0
	+	10 maq001	OP00123	2025-06-13 21:52:49	2	0
	478					

SQLITE ... v0.10.6

1

Page 1 / 5

Copy as TSV

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

python - src

Exportações em CSV e JSON

The screenshot shows the VS Code interface with the Explorer sidebar on the left. The 'output' folder is expanded, showing 'leituras.csv' selected. The main editor displays the contents of 'leituras.csv' in a terminal window. The file contains a CSV with 7 columns: id, codMaquina, ordemProducao, dataHora, distancia, folhas, and data. It lists 28 rows of sensor data.

```
python > src > output > leituras.csv > data
1 id;codMaquina;ordemProducao;dataHora;distancia;folhas;data
2 1;maq001;OP00123;2025-06-13 21:52:44;2.0;0;2025-06-13
3 2;maq001;OP00123;2025-06-13 21:52:45;2.0;0;2025-06-13
4 3;maq001;OP00123;2025-06-13 21:52:45;2.0;0;2025-06-13
5 4;maq001;OP00123;2025-06-13 21:52:46;2.0;0;2025-06-13
6 5;maq001;OP00123;2025-06-13 21:52:46;2.0;0;2025-06-13
7 6;maq001;OP00123;2025-06-13 21:52:47;2.0;0;2025-06-13
8 7;maq001;OP00123;2025-06-13 21:52:47;2.0;0;2025-06-13
9 8;maq001;OP00123;2025-06-13 21:52:48;2.0;0;2025-06-13
10 9;maq001;OP00123;2025-06-13 21:52:48;2.0;0;2025-06-13
11 10;maq001;OP00123;2025-06-13 21:52:49;2.0;0;2025-06-13
12 11;maq001;OP00123;2025-06-13 21:52:49;2.0;0;2025-06-13
13 12;maq001;OP00123;2025-06-13 21:52:50;2.0;0;2025-06-13
14 13;maq001;OP00123;2025-06-13 21:52:50;2.0;0;2025-06-13
15 14;maq001;OP00123;2025-06-13 21:52:51;2.0;0;2025-06-13
16 15;maq001;OP00123;2025-06-13 21:52:51;2.0;0;2025-06-13
17 16;maq001;OP00123;2025-06-13 21:52:52;2.0;0;2025-06-13
18 17;maq001;OP00123;2025-06-13 21:52:52;2.0;0;2025-06-13
19 18;maq001;OP00123;2025-06-13 21:52:53;2.0;0;2025-06-13
20 19;maq001;OP00123;2025-06-13 21:52:53;2.0;0;2025-06-13
21 20;maq001;OP00123;2025-06-13 21:52:54;53.0;0;2025-06-13
22 21;maq001;OP00123;2025-06-13 21:52:54;194.0;0;2025-06-13
23 22;maq001;OP00123;2025-06-13 21:52:55;340.9;0;2025-06-13
24 23;maq001;OP00123;2025-06-13 21:52:55;397.9;0;2025-06-13
25 24;maq001;OP00123;2025-06-13 21:52:56;278.0;0;2025-06-13
26 25;maq001;OP00123;2025-06-13 21:52:56;119.9;0;2025-06-13
27 26;maq001;OP00123;2025-06-13 21:52:57;46.0;0;2025-06-13
28 27;maq001;OP00123;2025-06-13 21:52:57;2.0;1;2025-06-13
```

The screenshot shows the VS Code interface with the 'output' folder expanded, showing 'leituras.json' selected. The main editor displays the contents of 'leituras.json' in a terminal window. The file contains a JSON array with 3 objects, each representing a sensor reading with fields: id, codMaquina, ordemProducao, dataHora, distancia, folhas, and data.

```
python > src > output > leituras.json > {} 1
1 [
2   {
3     "id":1,
4     "codMaquina":"maq001",
5     "ordemProducao":"OP00123",
6     "dataHora":"2025-06-13T21:52:44.000",
7     "distancia":2.0,
8     "folhas":0,
9     "data":"2025-06-13T00:00:00.000"
10  },
11  [
12    {
13      "id":2,
14      "codMaquina":"maq001",
15      "ordemProducao":"OP00123",
16      "dataHora":"2025-06-13T21:52:45.000",
17      "distancia":2.0,
18      "folhas":0,
19      "data":"2025-06-13T00:00:00.000"
20    },
21    {
22      "id":3,
23      "codMaquina":"maq001",
24      "ordemProducao":"OP00123",
25      "dataHora":"2025-06-13T21:52:45.000",
26      "distancia":2.0,
27      "folhas":0,
28      "data":"2025-06-13T00:00:00.000"
29    }
30  ]
31 ]
```

Menu de opções para Análise dos dados coletados

```
SPRINT_2
> docs
> ESP32_Firmware
✓ python
  > assets
  ✓ database
    enfesto.db M
  ✓ src
    > __pycache__
    ✓ output
      leituras.csv U
      leituras.json U
      analise_dados.py M
      db_manager.py
      main.py
      monitorar_sensor.py M
  ✓ venv
    > Include
    > Lib
    > Scripts
    > share
    .gitignore
    pyvenv.cfg
    README.md M
    requirements.txt

python > src > output > {} leituras.json > {} 1
1 [
2   {
3     "id":1,
4     "codMaquina":"maq001"

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Leitura inserida com sucesso: maq002 | OP=OP00221 | 399.9cm | folhas=0
Conexão com banco encerrada.
2025-06-13 23:30:26,191 [INFO]
Monitoramento encerrado pelo usuário.
(venv) PS D:\fiap\fase_4\Sprint_2\python\src> python .\main.py
✓ Tabela 'leituras' verificada/criada com sucesso.
Banco conectado em: ../database/enfesto.db

=== MENU PRINCIPAL ===
1 - Exibir análise: folhas por ordem de produção
2 - Exibir análise: produtividade por máquina
3 - Exibir análise: folhas por dia
4 - Exportar CSV/JSON
5 - Visualizar gráfico: folhas por ordem
6 - Visualizar gráfico: produtividade por máquina
7 - Visualizar gráfico: folhas por dia
0 - Sair
Escolha uma opção: 
```

1 - Exibir análise: folhas por ordem de produção

```
Escolha uma opção: 1
```

ordemProducao	codMaquina	inicio	fim	total_folhas
OP00123	maq001	2025-06-13 21:52:44	2025-06-13 22:57:10	8
OP00124	maq001	2025-06-13 22:57:57	2025-06-13 22:58:27	7
OP00224	maq002	2025-06-13 23:13:08	2025-06-13 23:13:53	6
OP00221	maq002	2025-06-13 23:14:18	2025-06-13 23:27:04	7

2 - Exibir análise: produtividade por máquina

```
Escolha uma opção: 2
```

codMaquina	folhas	tempo_horas	folhas_por_hora
maq001	15	1.082222	13.860370
maq002	13	0.225278	57.706535

3 - Exibir análise: folhas por dia

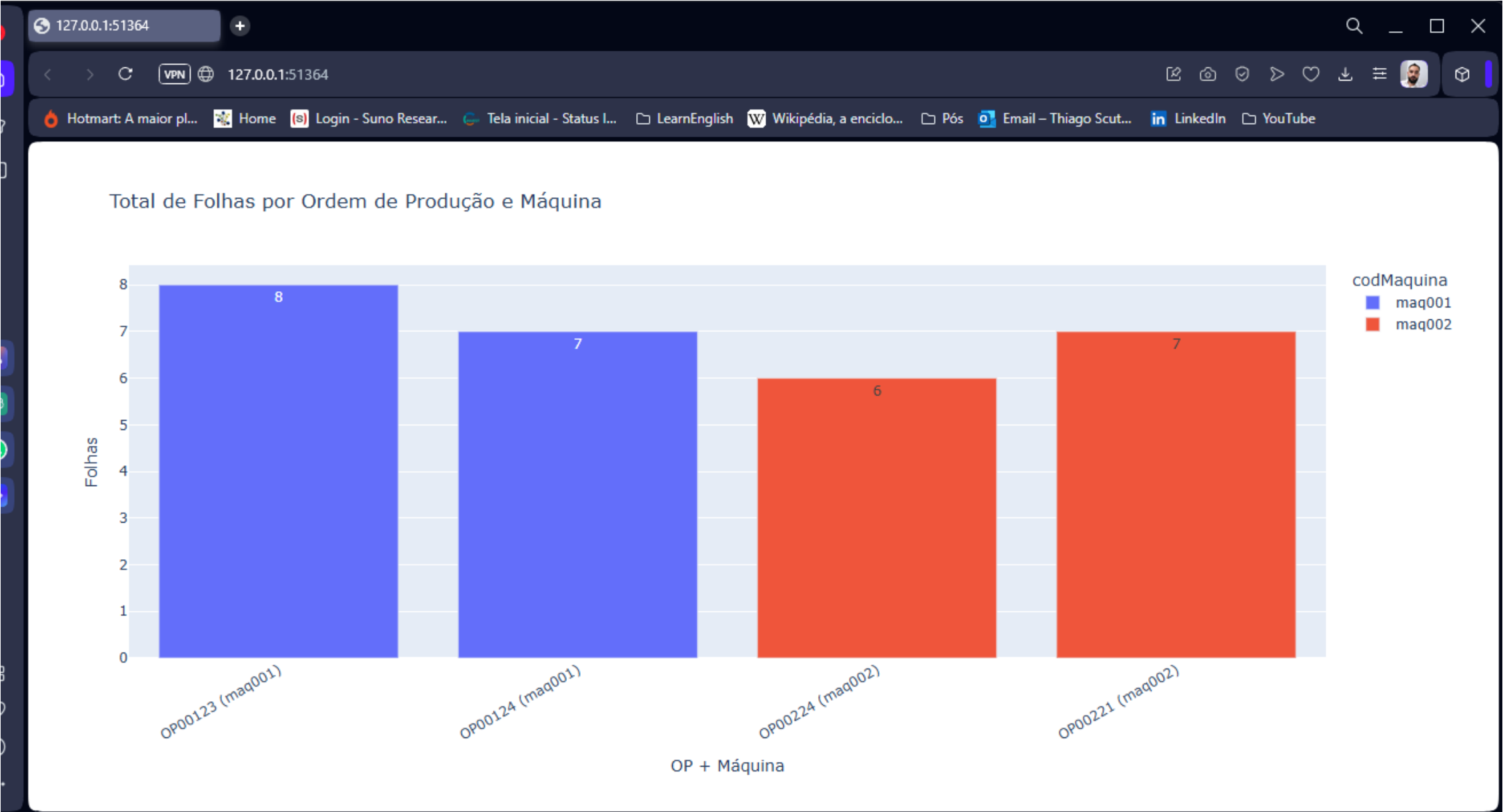
```
Escolha uma opção: 3
```

data	codMaquina	ordemProducao	folhas
2025-06-13	maq001	OP00123	8
2025-06-13	maq001	OP00124	7
2025-06-13	maq002	OP00221	7
2025-06-13	maq002	OP00224	6

4 - Exportar CSV/JSON

```
Escolha uma opção: 4
2025-06-13 23:33:13,765 [INFO] CSV exportado com sucesso: output\leituras.csv
2025-06-13 23:33:13,766 [INFO] JSON exportado com sucesso: output\leituras.json
```

5 - Visualizar gráfico: folhas por ordem



6 - Visualizar gráfico: produtividade por máquina



6 - Visualizar gráfico: produtividade por máquina

