



Projeto 04

Coisas à Venda – Prática

Jan K. S. – janks@puc-rio.br

ENG4051 – Projeto Internet das Coisas

Testes Iniciais

Tabela Produtos

número inteiro (long long)	texto	número inteiro (int)
codigo integer ▲	nome varchar(255) ▲	preco integer ▲
78900000000062	Concha Mini Funda Linux	199
78900000000116	Saco Lixo Bio Lixo 100l	5999
78900000000123	Ferro De Solda Estanhador 12v Estanhadores Brasi...	9499
78900000000543	Base Misturador Monocomando Com Acabamento...	8199
78900000000604	Posta De Cacao Azul S pele Iglu 1kg	2399 → R\$ 23.99
78900000000826	Cerveja Farroupilha Vienna Lager Pessego	5399
78900000001236	Cuba De Apoio Para Banheiro Proaqua Com Ladrã...	1999
78900000001243	Empada Trigo E Salsa Frango	7199
78900000001441	Empanado De Calabresa	7599
78900000001458	Insumos Lombo Cozido Em Cubos Kg	9099
78900000001762	Empada Trigo E Salsa Palmito	5699

Tabela Produtos Usada no PostgreSQL e no SQLite

1

No ESP32, exiba no display o código de barras para "7891528038810" e o QR Code para o link "https://janks.link/produto".

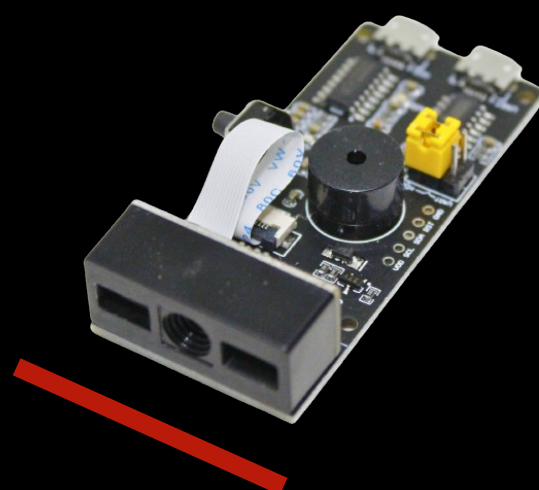


2

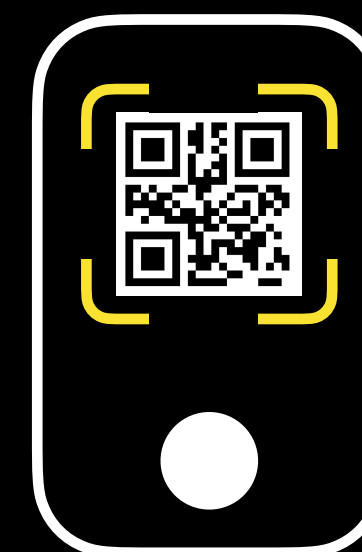
Inicialize o leitor de código de barras e leia o código desenhado no display.

Use o código para buscar os dados do produto na tabela produtos do banco de dados. Imprima os campos "nome" e o "preço" do produto da Serial.

Abra o QR Code no seu celular via câmera.



Nome do Produto
R\$ 19.99



3

No Node-RED, ao receber dígitos num tópico "codigo-de-barras", busque no PostgreSQL na tabela "produtos" com a coluna "codigo" e imprima o resultado com o Debug.

Teste com o MQTTX com alguns códigos.



Testes Iniciais

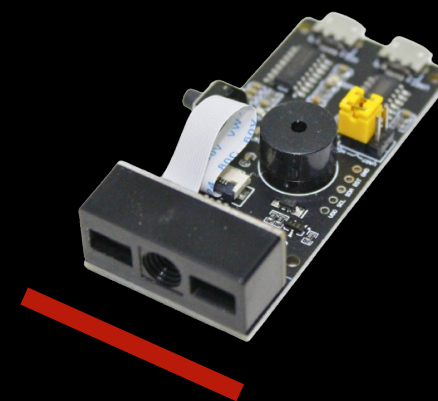
Protótipo



Caixa de Supermercado com Auto Atendimento

1

Ao iniciar o programa, exiba o texto no display com a mensagem de início, ativando o leitor para leitura contínua

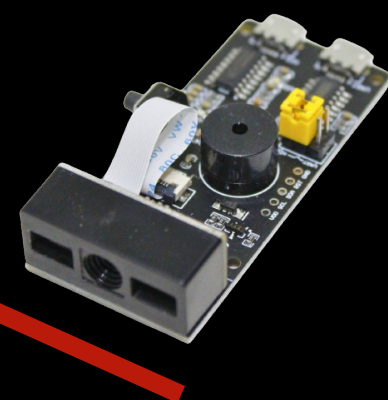


Bem-vindo!

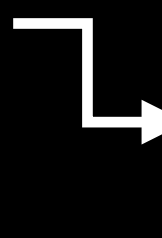
Passe o produto no leitor

2

Ao escanear um código, procure-o na tabela produtos do SQLite. Se encontrar, adicione seu dicionário numa lista JSON global e some o preço numa variável global `totalDaCompra` tipo `int`.



SQLite



```
[
  {"nome": "Água", "preço": 699, ...},
  ...
  {"nome": "Twix", "preço": 2399, ...}
]
```

`totalDaCompra = 9576`

3

Após adicionar o dicionário, percorra os últimos 5 itens da lista e desenhe os primeiros 25 caracteres do nome e o preço de cada um no display.
 ↳ DICA: ajuste o índice inicial do `for` se o tamanho da lista for maior que 5. Use a fonte `u8g2_font_helvR12_te` (regular 12) e espaçamento de 18px entre itens. Teste primeiro no simulador em janks.link/iot/tela.

Água Mineral Minalba	R\$ 6.99
Pasta de Dente Tandy	R\$ 9.99
Iogurte Grego Vigor	R\$ 4.69
Suco de Uva Bendita	R\$ 24.99
Twix	R\$ 4.79

Protótipo: Escaneamento e Listagem de Itens

4

No canto esquerdo inferior, desenhe o código de barras do produto atual com escala igual a 1.

No canto direito inferior, desenhe o valor total da compra.

Água Mineral Minalba	R\$ 6.99
Pasta de Dente Tandy	R\$ 9.99
Iogurte Grego Vigor	R\$ 4.69
Suco de Uva Bendita	R\$ 24.99
Twix	R\$ 4.79
	
7 891528 038810	
R\$ 51.49	

5

Ajuste as variáveis globais Pix com os dados da sua conta bancária.

Ao apertar o botão central do joystick (pino 5), desligue o leitor de código e desenhe na tela o QR Code de Pix com o total da compra.

Teste abrir o código no aplicativo do seu banco.

↪ DICA: NÃO FINALIZE A TRANSFERÊNCIA PARA OUTRA PESSOA!



Protótipo: Total e Pagamento em Pix

Integração

Tabela Vendas

id
(automático)

data/hora
da venda

código do
produto
vendido

preço da
venda

	id int4(32,0) ▲	data_hora timestam... ▲	codigo int8(64,0) ▲	preco int4(32,0) ▲
1	40	2024-09-18 20:01:54...	7894900530209	4699
2	41	2024-09-18 20:01:54...	7894900530209	4699
3	42	2024-09-19 20:42:24...	7891098010575	9699
4	43	2024-09-19 20:42:24...	7891098010575	9699
5	44	2025-04-04 19:18:42...	7891098010575	9600
6	45	2025-04-04 19:19:20...	7891098010575	9600
7	46	2025-04-04 19:20:12...	7891098010575	9600
8	159	2025-04-07 18:43:33...	7896094209701	9299
9	160	2025-04-07 18:44:04...	7896094209701	9299
10	161	2025-04-07 18:44:59...	7896094209701	9299
11	162	2025-04-07 18:47:12...	7896094209701	9299

Tabela de Vendas de Todos os Caixas

1

No ESP32, ao receber o tópico "pix-confirmado/NUMERO_DA_SUA_BANCADA", envie o JSON das compras ao tópico "compra/NUMERO_DA_SUA_BANCADA".

Reinicie também as variáveis globais `totalDaCompra` e `itensComprados` e mostre o texto "Caixa Livre" no display.

compra/NUMERO_DA_SUA_BANCADA

Teste com o MQTTX.

```
[
  {"codigo": "7896065880069", "preço": 699, ...},
  {"codigo": "Twix", "preço": 2399, ...}, ...
]
```

2

No ESP32, publique o número da sua bancada no tópico "caixa-ocupado" ao se conectar com o MQTT no começo do programa.

Publique o número no tópico "caixa-livre" ao receber a confirmação do Pix.

caixa-ocupado

NUMERO_DA_SUA_BANCADA

caixa-livre

NUMERO_DA_SUA_BANCADA

3

No Node-RED, ao receber o tópico de compra da sua bancada, use o bloco Split para quebrar a lista em várias mensagens.

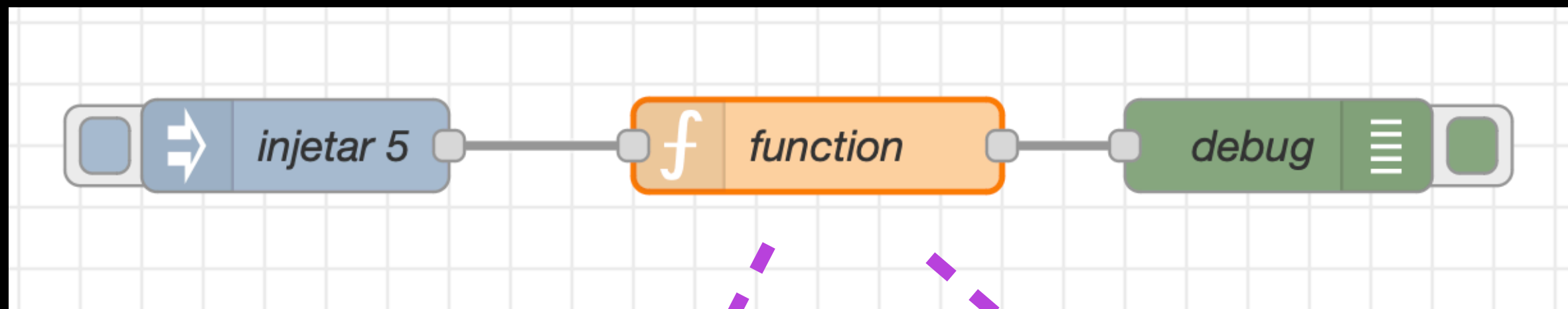
Use então o bloco Change para criar o campo `queryParameter` e depois inserir nele a data/hora atual num novo campo.

Enfim, use o bloco PostgreSQL para inserir um item por vez na tabela vendas. Verifique o resultado no Beekeeper Studio.



Integração: Adição de Itens Vendidos no Banco de Dados

Desafio Extra



13/09/2024, 15:20:44 nó: debug 26
msg.payload : array[1]
▶ ["Jan"]

13/09/2024, 15:20:44 nó: debug 26
msg.payload : array[2]
▶ ["Jan", "Jan"]

13/09/2024, 15:20:44 nó: debug 26
msg.payload : array[3]
▶ ["Jan", "Jan", "Jan"]

⚙️ Propriedades

Nome function 6

⚙️ Configurar No início Na mensagem

```
1 // 0 código adicionado aqui será executado
2 // sempre que o nó for iniciado.
3 flow.set("minhaLista", []);
```

⚙️ Propriedades

Nome function 6

⚙️ Configurar No início Na mensagem

```
1 var lista = flow.get("minhaLista");
2 lista.push("Jan");
3 flow.set("minhaLista", lista);
4
5 msg.payload = lista;
6 return msg;
```

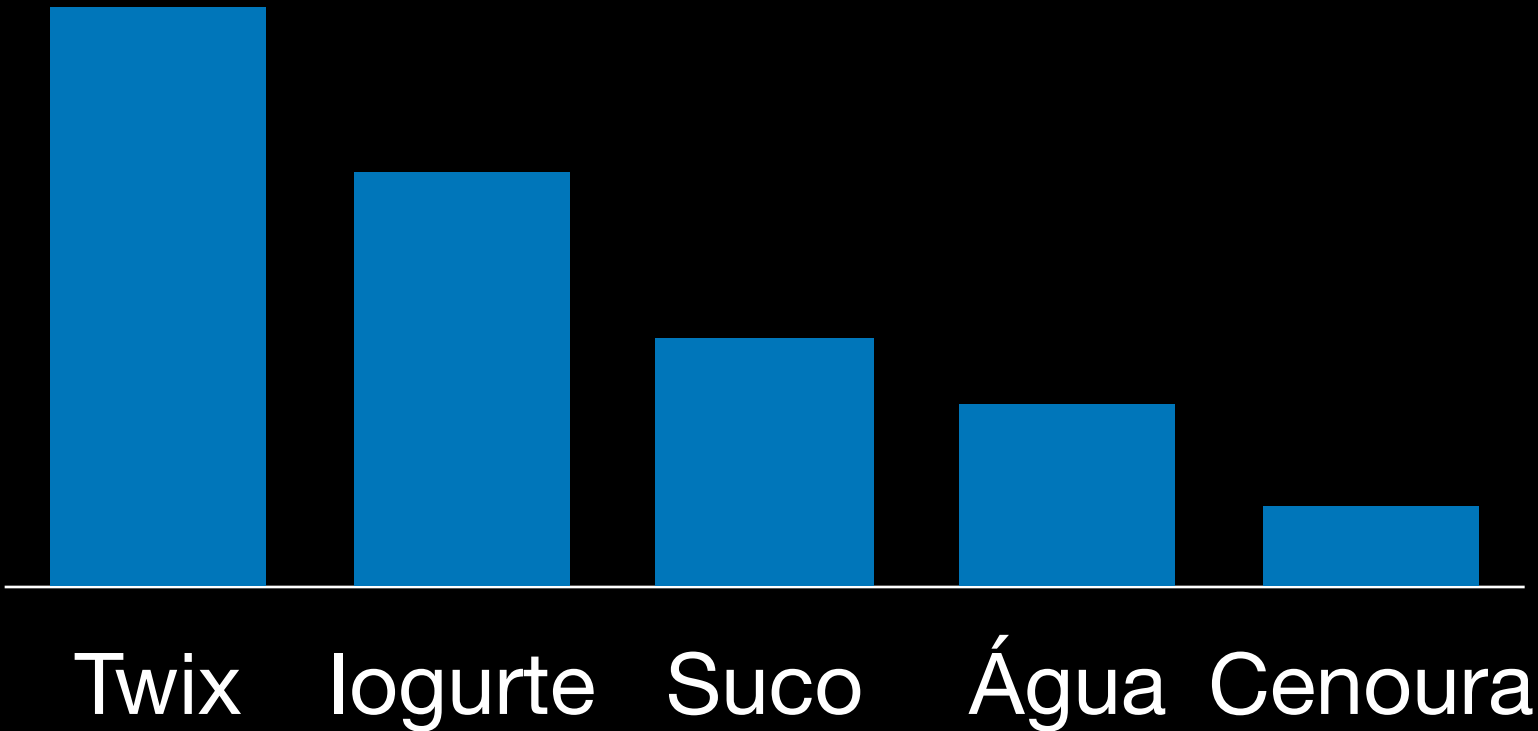


A valor na variável **flow** fica persistente para todos os nós do flow ao longo da execução.

Bloco Function com Variável Persistente

1

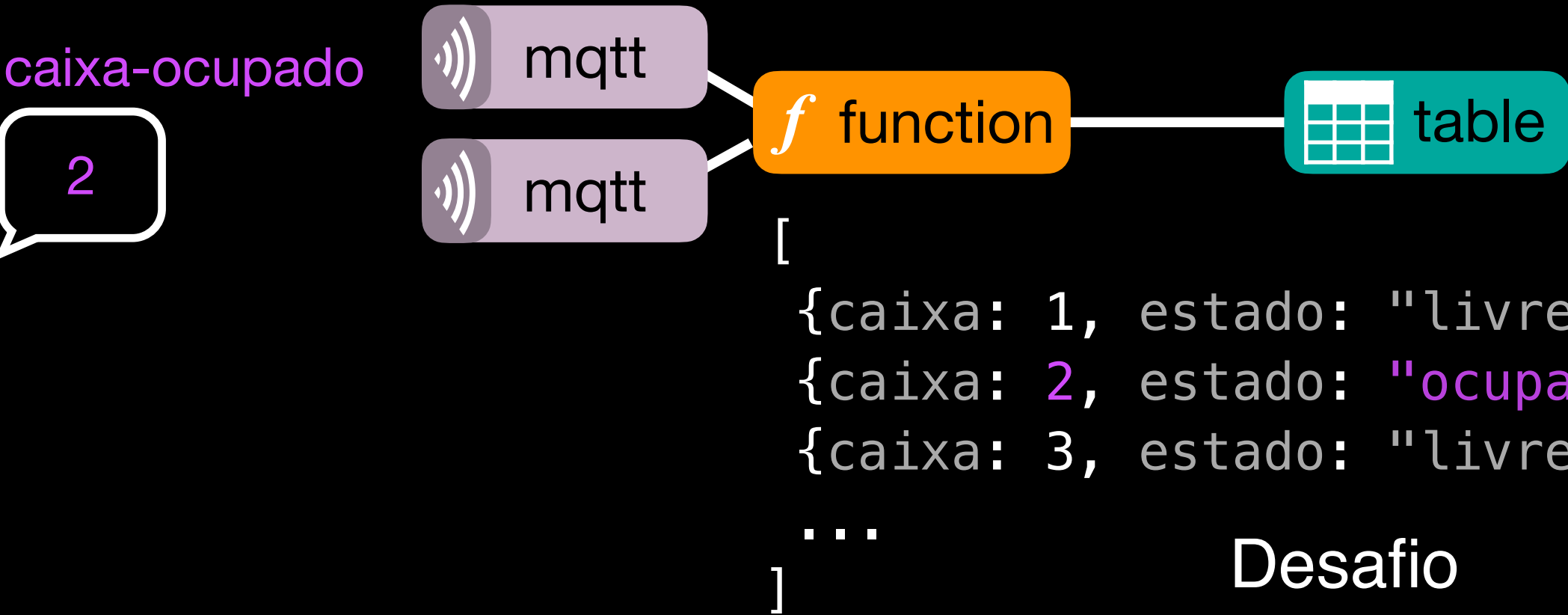
No Node-RED, use o bloco inject para ativar a cada 5 segundos. Faça um gráfico de barras com o total de itens vendidos de cada produto, ordenando por esse total em ordem decrescente e limitando a 5 itens. ↪ DICA: pesquise sobre como usar o COUNT(*), GROUP BY e o INNER JOIN no SQL. Use também o ORDER BY e o LIMIT.



2

No Node-RED, ao receber os tópicos "caixa-livre" ou "caixa-ocupado", use o bloco Function para atualizar uma lista de dicionários com o estado dos caixas dentro do flow.

Passe essa lista para um Table do Dashboard.



Caixa	Estado
1	livre
2	ocupado
3	livre
4	livre

...



janks.link/iot/projeto04.zip

Material do Projeto 04