



# Projeto 05

## Coisas Consumíveis – Teoria

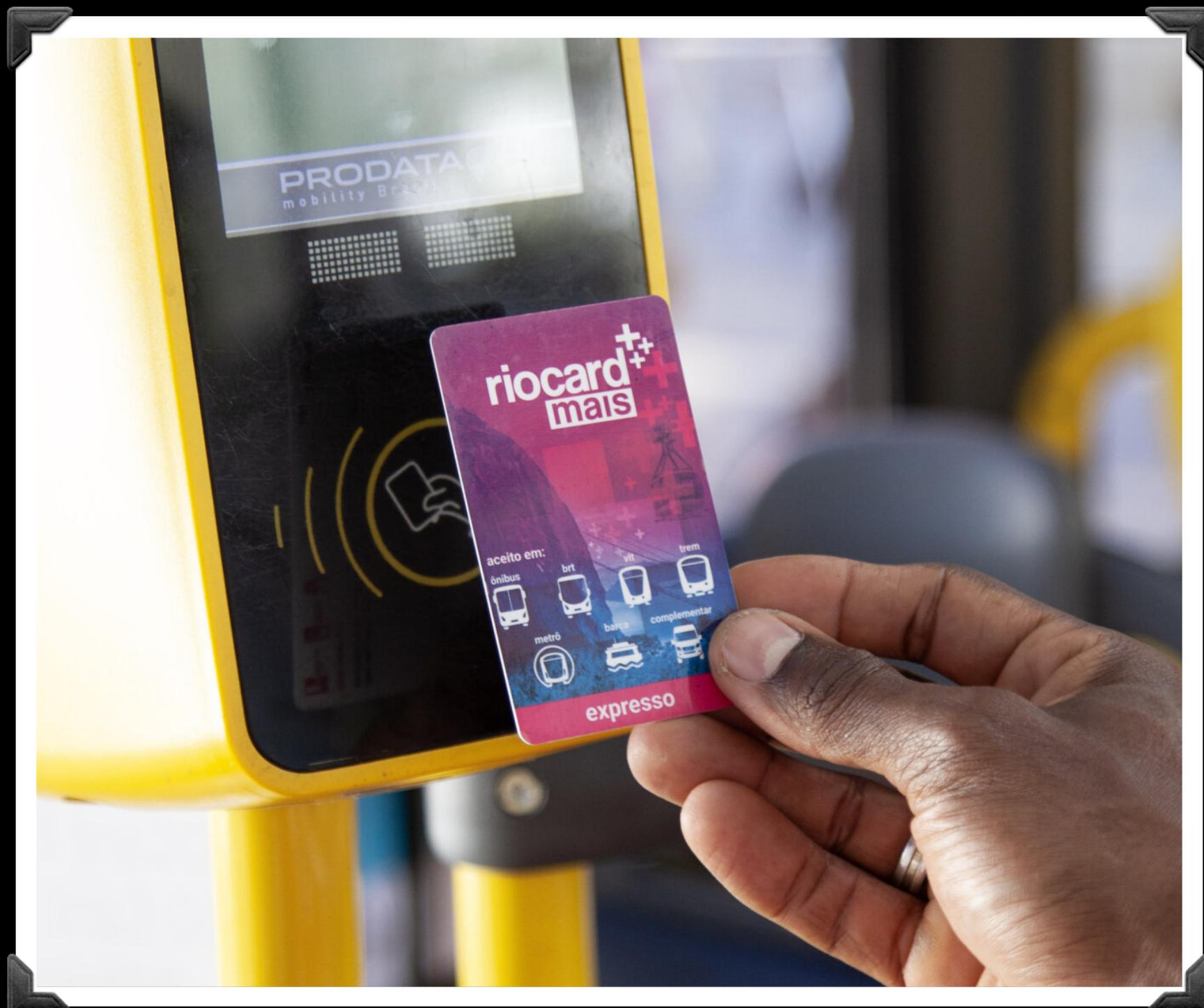
Jan K. S. – [janks@puc-rio.br](mailto:janks@puc-rio.br)

ENG4051 – Projeto Internet das Coisas

Coisas



Leitor RFID e Cartões MIFARE Classic

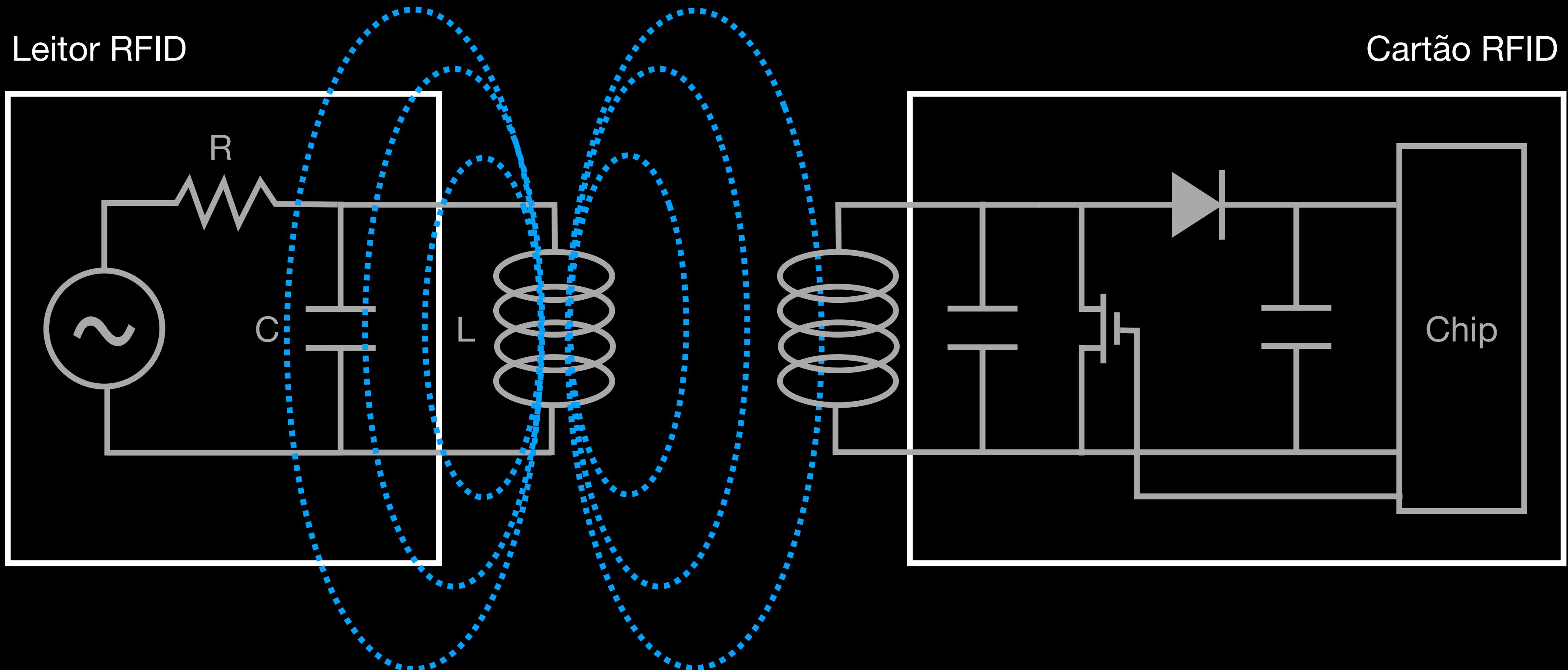


Exemplos de Aplicações do RFId

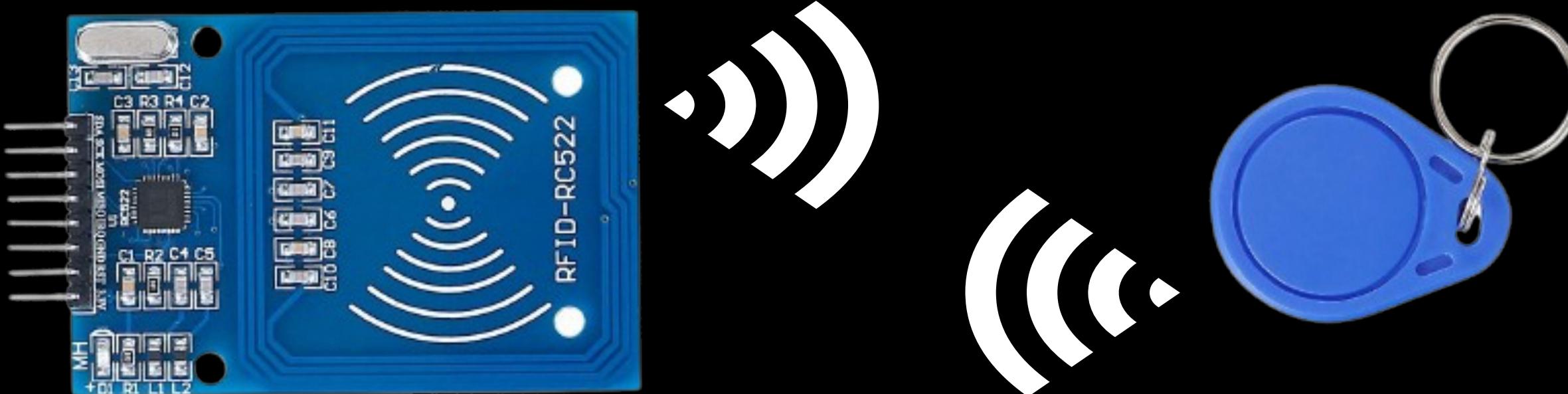
campo magnético

Leitor RFID

Cartão RFID



Acorda aí! Qual o seu id?



E7 45 D6 19

Funcionamento do RFId

## setor do RFID

	condições de acesso										chave secreta B				
	chave secreta A														
bloco trailer	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF
bloco de dado	50	55	43	2D	52	69	6F	00	00	00	00	00	00	00	PUC–Rio
bloco de dado	4A	61	6E	20	4B	2E	20	53	2E	00	00	00	00	00	Jan K. S.
bloco de dado	49	6F	54	00	00	00	00	00	00	00	00	00	00	00	IoT

O acesso aos dados no setor exige saber a chave A ou a B.

Elas podem ter permissões específicas diferentes para ler e modificar os dados.

Mas existe uma **falta de segurança** nesse sistema, então **não guarde dados sensíveis dentro do RFID**.



Setor	Bloco	Dados															
15	63	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
	62	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	61	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	...	...							...								
2	11	A7	A8	A9	AA	AB	AC	FF	07	80	69	B7	B8	B9	BA	BB	BC
	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1	7	FF	FF	FF	FF	FF	FF	FF	07	80	69	FF	FF	FF	FF	FF	FF
	6	50	55	43	2D	52	69	6F	00	00	00	00	00	00	00	00	00
	5	4A	61	6E	20	4B	2E	20	53	2E	00	00	00	00	00	00	00
	4	49	6F	54	00	00	00	00	00	00	00	00	00	00	00	00	00
0	3	A1	A2	A3	A4	A5	A6	FF	07	80	69	B1	B2	B3	B4	B5	B6
	2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	0	E7	45	D6	19	6D	08	04	00	62	63	64	65	66	67	68	69

(uid) id único

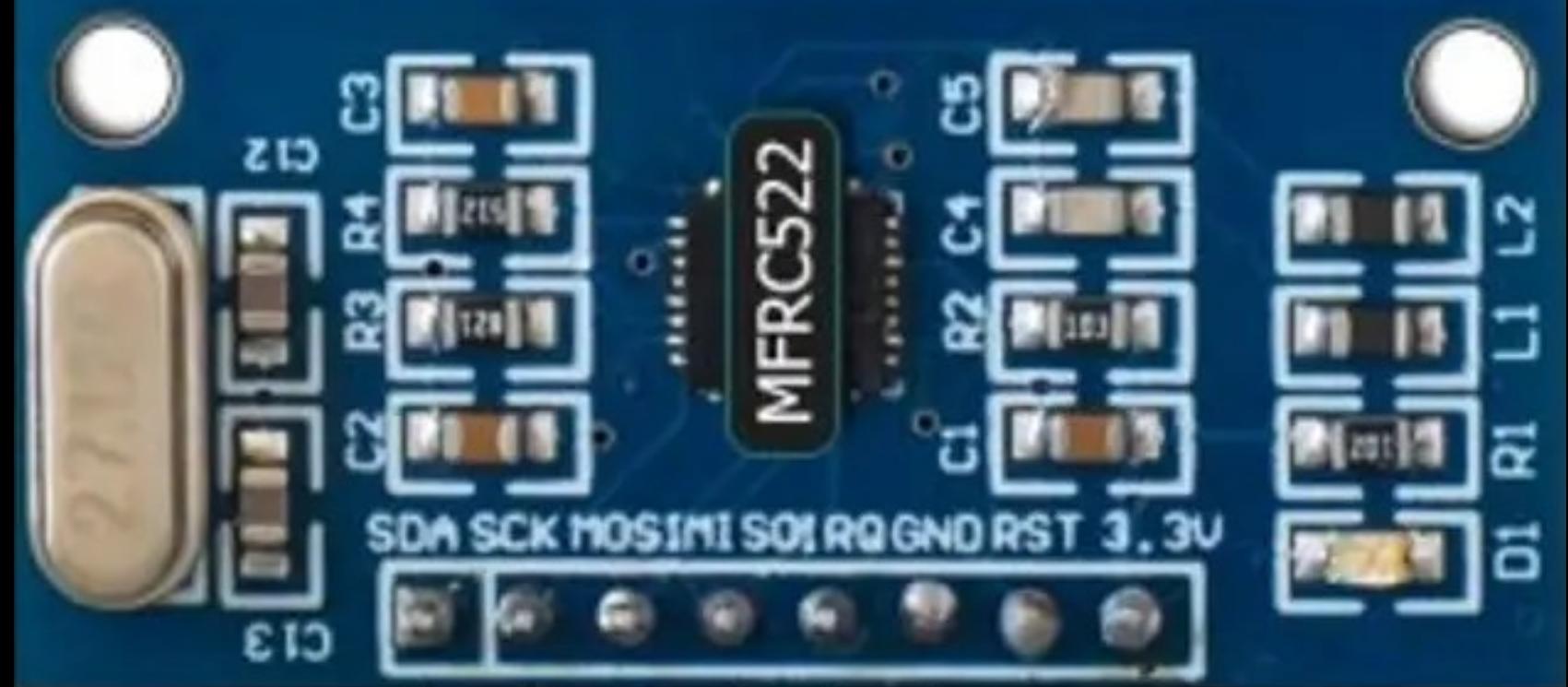
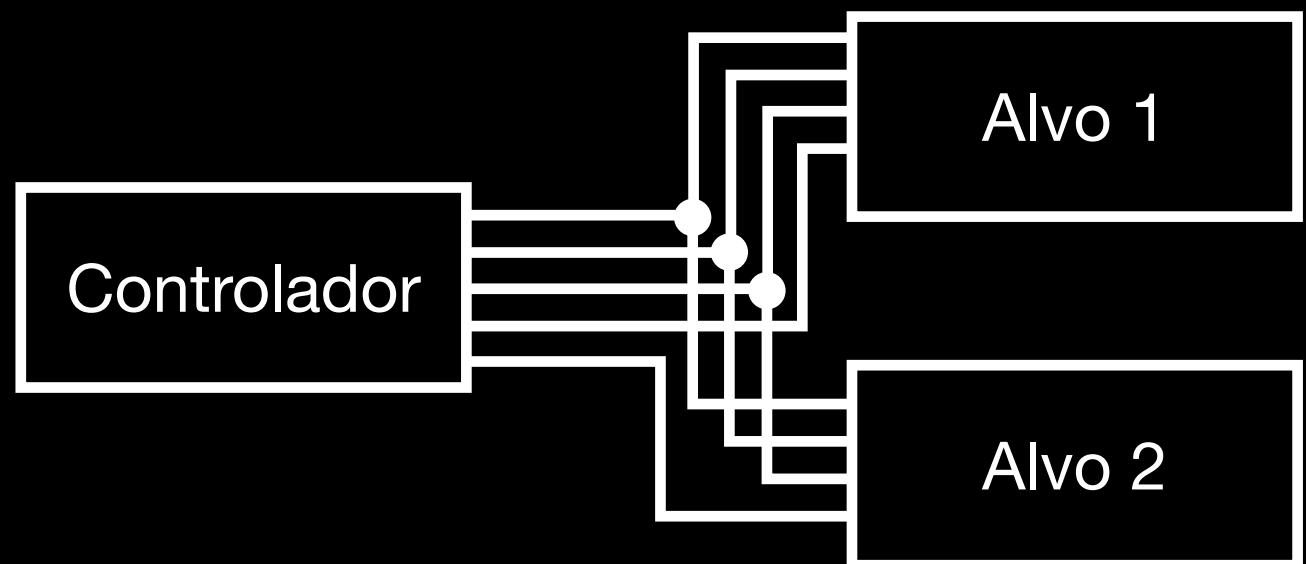
dados do fabricante

O bloco 0 geralmente  
é somente leitura.



Conjunto de Setores de Blocos do MIFARE

## SPI (Serial Peripheral Interface)



**Pino Leitor RFID MFRC522**

3.3V	3.3V
GND	GND
MISO	13
MOSI	11
SCK (Clock)	12
SDA (Chip Select)	46
RST (Reset)	17
IRQ	--

**Pino ESP32 S3 CAM**

} pinos padrão  
do SPI no  
ESP32 S3 CAM

Conexão do Leitor RFID no ESP32 S3 CAM

The image shows a GitHub repository page for the MFRC522 library and an open Arduino IDE window.

**GitHub Repository Page:**

- Header:** Shows the URL [github.com/miguelbalboa/rfid](https://github.com/miguelbalboa/rfid).
- README:** Contains links to [README](#) and [Unlicense license](#).
- Title:** **MFRC522**.
- Metrics:** maintained no! (as of 2019), PlatformIO CI passing, C++ 11, release v1.4.12, ArduinoIDE >=1.6.10.
- Contributors:** Shows a grid of 10 contributors and a link to + 66 contributors.
- Description:** Arduino library for MFRC522 and other RFID RC522 based readers using SPI interface.
- Text:** Read and write different types of Radio-Frequency cards on your Arduino using a RC522 based reader. Serial Peripheral Interface (SPI) interface.
- Development:** The development by owner miguelbalboa has ended.
- Feature status:** complete freeze; no function or API updates; no extensions for other boards; no new examples.
- Code status:** partial freeze; just fixes/typos or documentation updates; no new examples.

**Arduino IDE Window:**

- Toolbar:** Shows the sketch name `sketch_apr11a` and Arduino IDE version 2.3.4.
- Sketch Editor:** Displays the code for `sketch_apr11a.ino`.

```
1 void setup() {  
2     // put your setup code here  
3 }  
4  
5 void loop() {  
6     // put your main code here  
7 }  
8  
9 }  
10 }
```
- Library Manager:** Shows the MFRC522 library installed at version 1.4.12. It includes a description: "Arduino RFID Library for MFRC522 (SPI) Read/Write a RFID Card or Tag using the ISO/IEC 14443A/MIFARE interface.", a "More info" link, and a "REMOVE" button.
- Status Bar:** Shows "Ln 1, Col 1" and "No board selected".

```
#include <SPI.h>
#include <MFRC522.h>

MFRC522 rfid(46, 17);
MFRC522::MIFARE_Key chaveA = {{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF}};

String lerUID() {
    String id = "";
    for (byte i = 0; i < rfid.uid.size; i++) {
        if (i > 0) {
            id += " "; // espaço depois do byte anterior
        }
        if (rfid.uid.uidByte[i] < 0x10) {
            id += "0"; // para ficar "07" em vez de "7", por exemplo
        }
        id += String(rfid.uid.uidByte[i], HEX);
    }
    id.toUpperCase();
    return id;
}
// código continua a seguir...
```

```
// continuação do código anterior...
```

```
String lerTextoDoBloco(byte bloco) {
    byte tamanhoDados = 18; // 16 bytes de dados + 2 para CRC
    char dados[tamanhoDados];

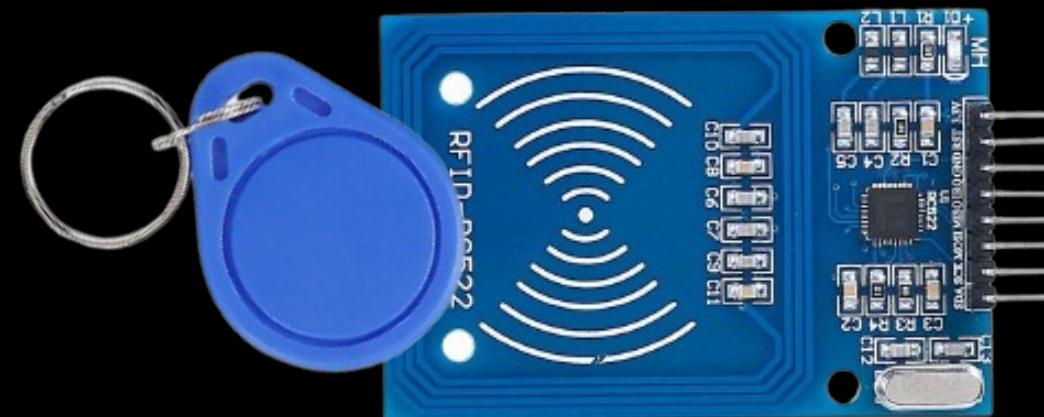
    MFRC522::StatusCode status = rfid.PCD_Authenticate(
        MFRC522::PICC_CMD_MF_AUTH_KEY_A, bloco, &chaveA, &(rfid.uid)
    );
    if (status != MFRC522::STATUS_OK) {
        return "";
    }

    status = rfid.MIFARE_Read(bloco, (byte*)dados, &tamanhoDados);
    if (status != MFRC522::STATUS_OK) {
        return "";
    }
    dados[tamanhoDados - 2] = '\0';
    return String(dados);
}

// código continua a seguir...
```

```
// continuação do código anterior...
```

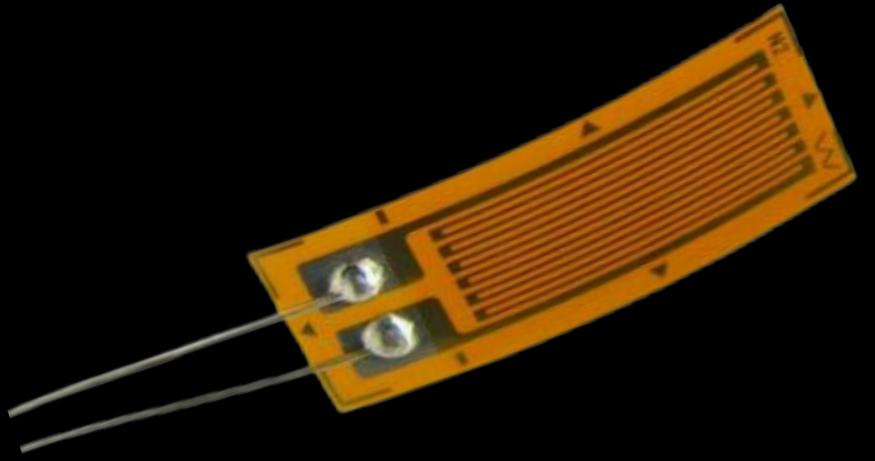
```
void setup() {  
    Serial.begin(115200); delay(500);  
  
    SPI.begin();  
    rfid.PCD_Init();  
}  
  
void loop() {  
    if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()){  
        String id = lerUID();  
        Serial.println("UID da tag: " + id);  
  
        String texto = lerTextoDoBloco(6);  
        Serial.println("Texto no bloco 6: " + texto);  
  
        rfid.PICC_HaltA(); // interrompe leitura (não fica repetindo)  
        rfid.PCD_StopCrypto1();  
    }  
}
```



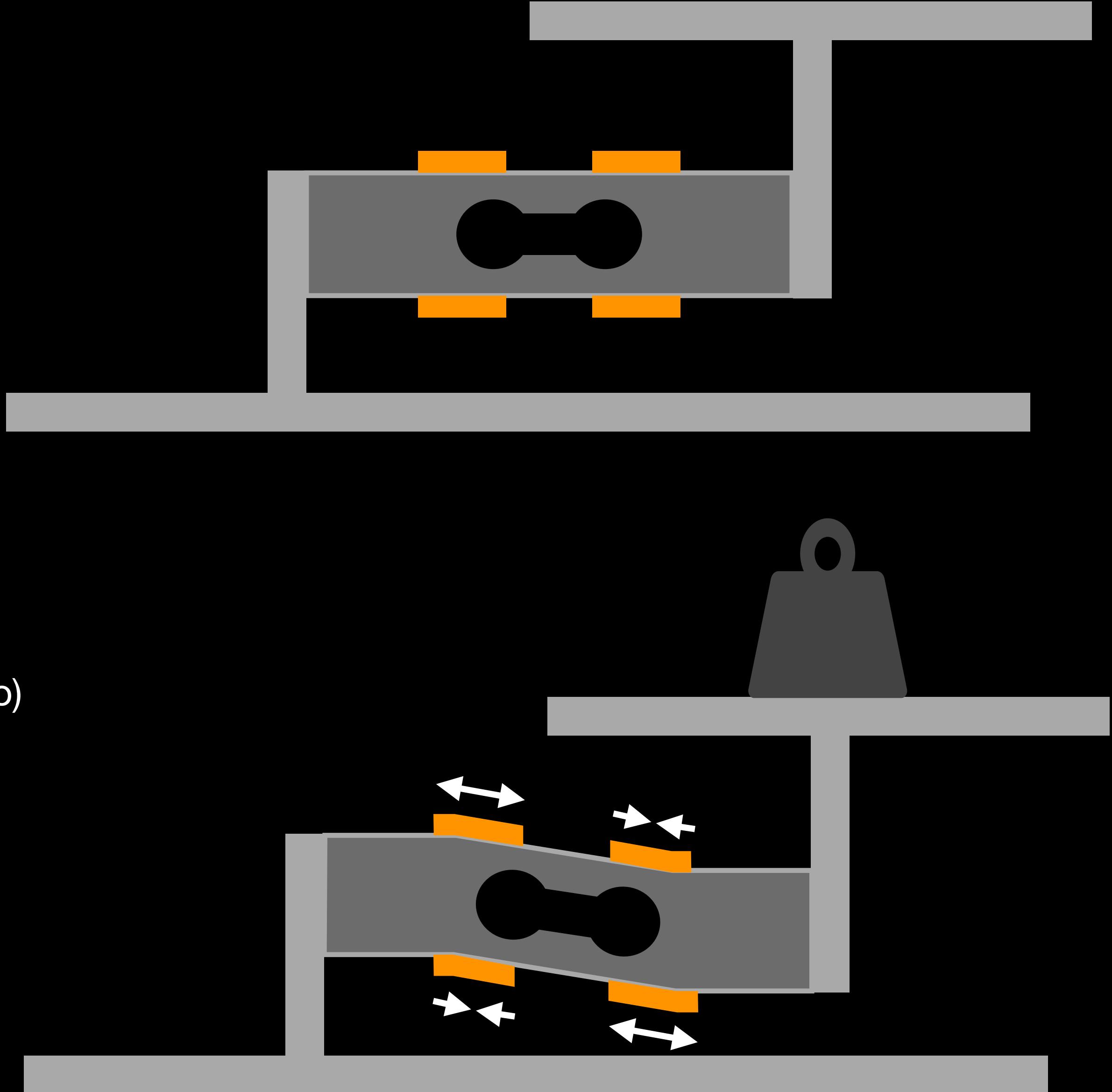
UID da tag: E7 45 D6 19  
Texto no bloco 6: PUC-Rio



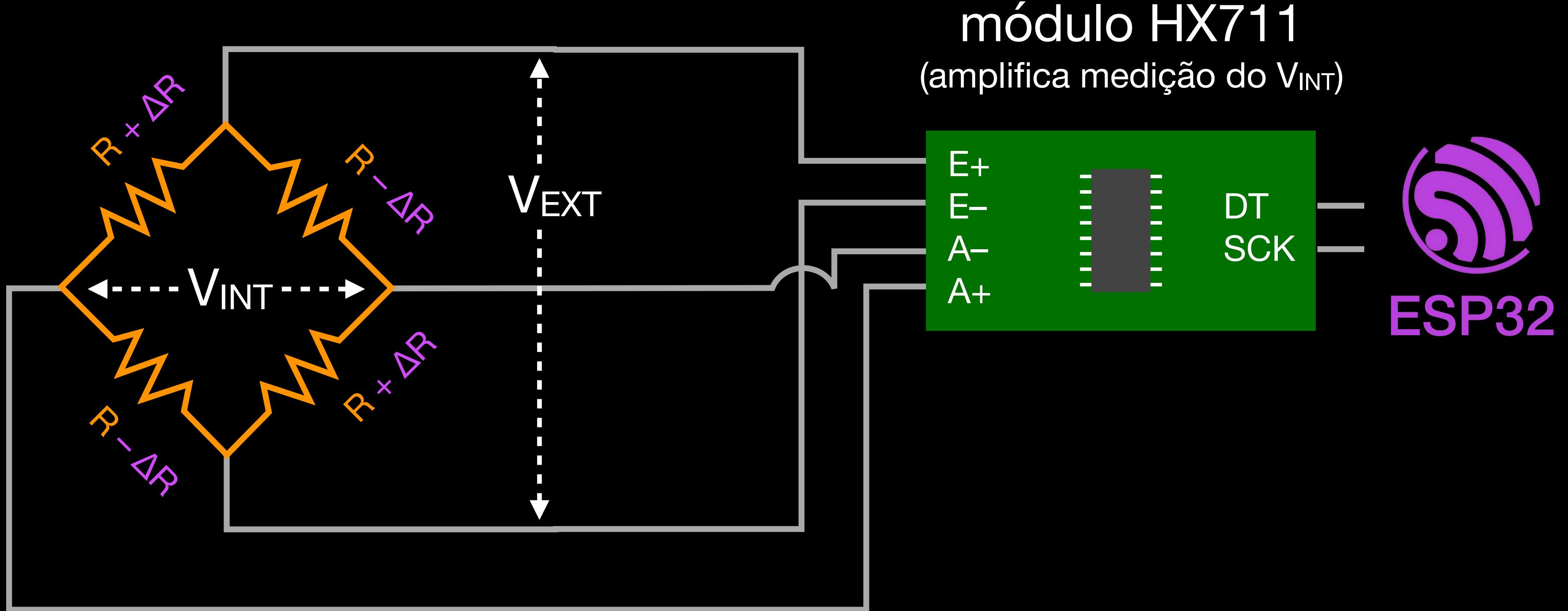
Balança com Célula de Carga



**extensômetro**  
(resistor variável com deformação)



Deformação de Extensômetros após Aplicação de Carga



$$V_{INT} = V_{EXT} \frac{\Delta R}{R}$$

Medição com Ponte de Wheatstone + Módulo Amplificador

github.com/bogde/HX711

# HX711

An Arduino library to interface the [Avia Semiconductor HX711 24-Bit Analog-to-Digital Converter \(ADC\)](#) for reading load scales.

It supports the architectures `atmelavr`, `espressif32`, `atmelsam`, `teensy` and `ststm32` [PlatformIO targets](#).

## Synopsis

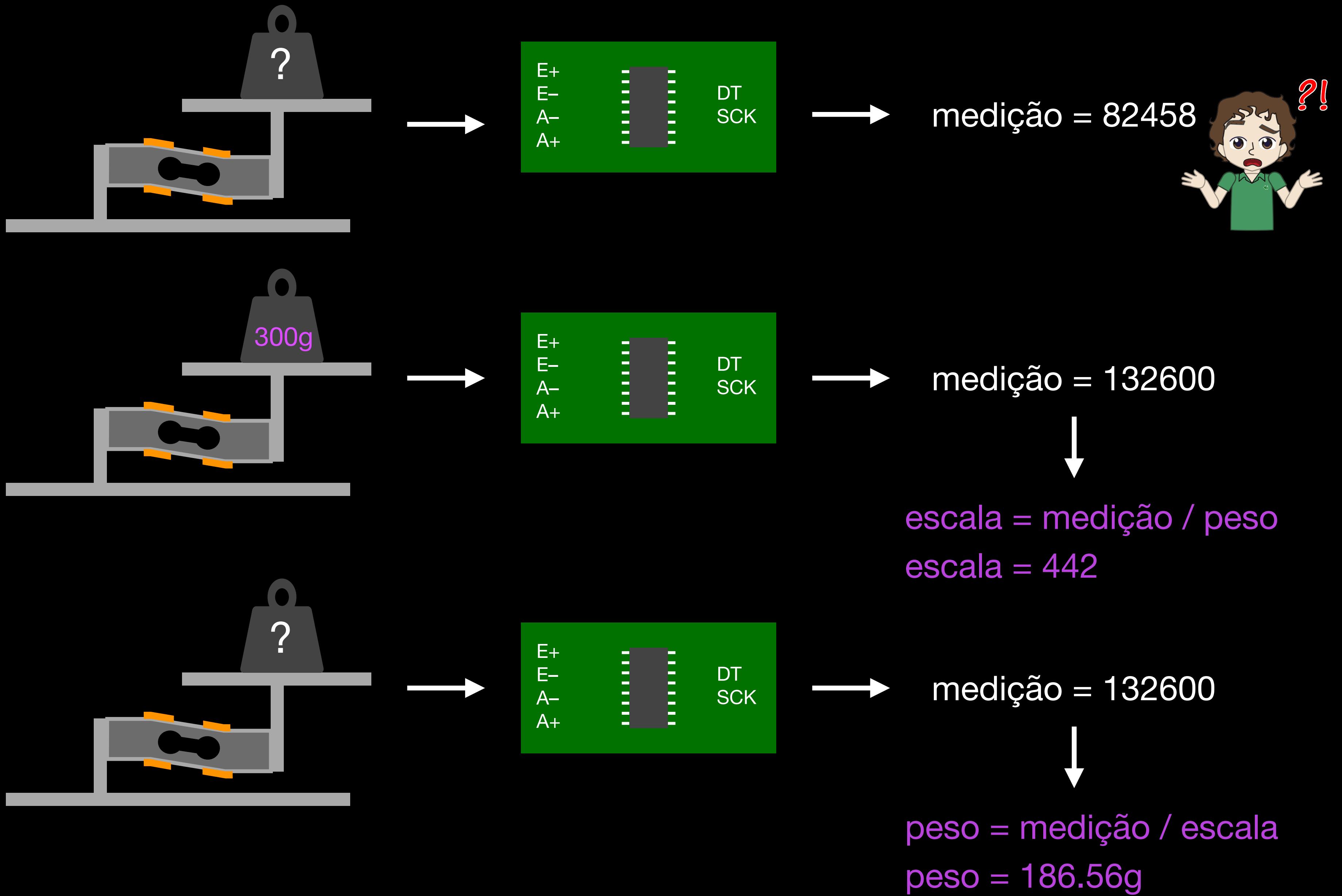
### Blocking mode

The library is usually used in blocking mode, i.e. it hardware becoming available before returning a re

```
#include "HX711.h"
HX711 loadcell;
```

The screenshot shows the Arduino IDE interface. At the top, there's a browser-like header with tabs for 'README' and 'MIT license'. Below the header, there's a grid of small profile pictures. To the right of the grid, the word 'Languages' is followed by 'sketch\_apr11a | Arduino IDE 2.3.4'. The main area of the IDE has a teal header bar with icons for saving, loading, and selecting a board. The 'Select Board' dropdown is open, showing options like 'ATmega328P (Arduino Uno)', 'ATmega32U4 (Arduino Leonardo)', and 'ESP32 Dev Module'. The central workspace contains two panes: a 'LIBRARY MANAGER' pane on the left and a 'sketch\_apr11a.ino' code editor on the right. The code editor shows a simple setup and loop function. The bottom status bar indicates 'Ln 1, Col 1' and 'No board selected'.

HX711: Biblioteca para Célula de Carga com Módulo HX711



Cálculo do Fator de Escala da Balança

```
#include <HX711.h>

HX711 balanca;

void setup() {
    balanca.begin(6, 7);      // pinos de comunicação
    balanca.set_scale(SEU_FATOR_DE_ESCALA); // escala de calibração
}                                // em alguma função do código...

// 1 medição (em gramas)
float pesoMedido = balanca.get_units(1);

// média de 5 medições seguidas (trava o programa)
float pesoMedio = balanca.get_units(5);

// define tara (peso inicial para ignorar) com 5 medições
balanca.tare(5);
```

Exemplo de Leitura da Célula de Carga

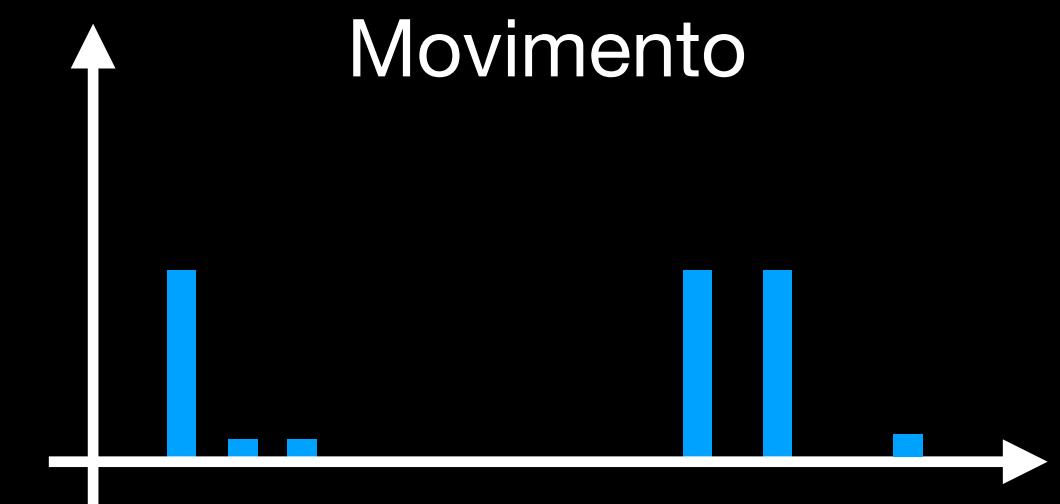
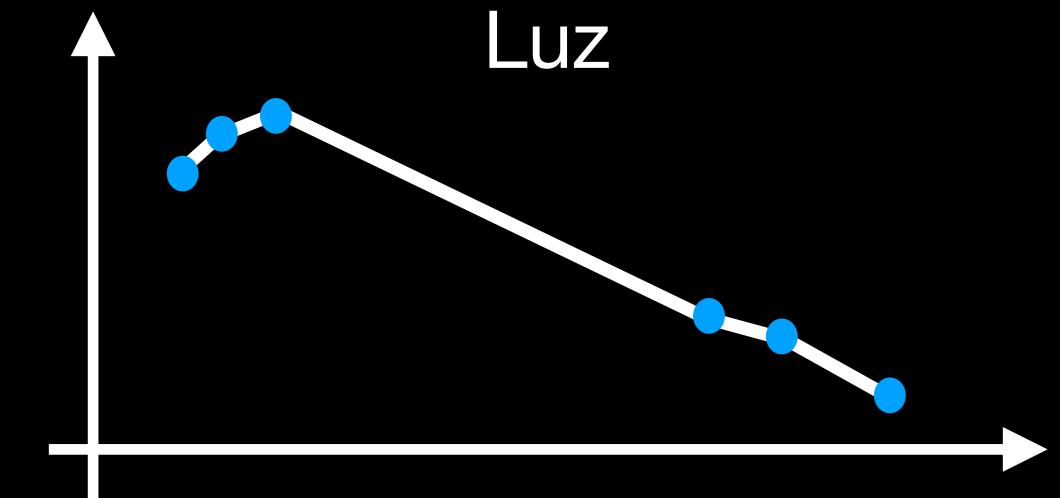
# Internet



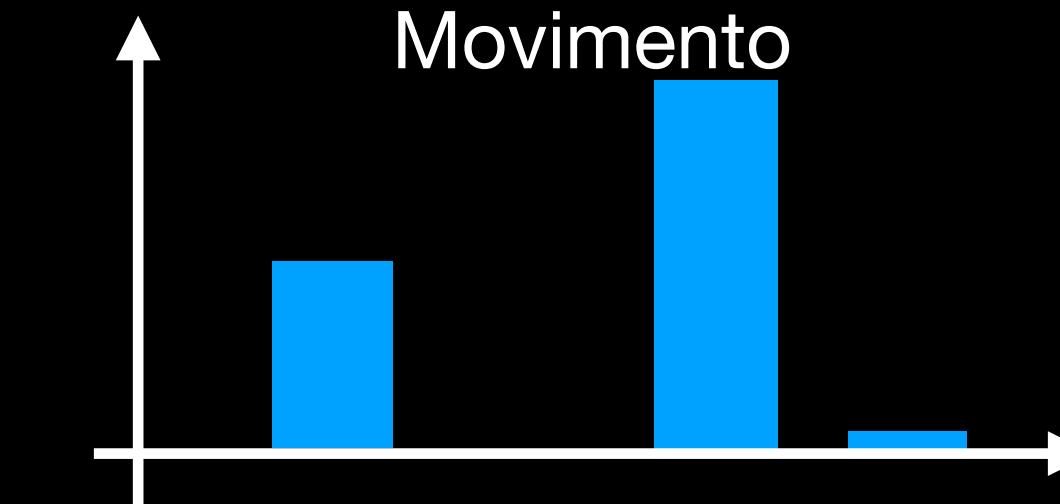
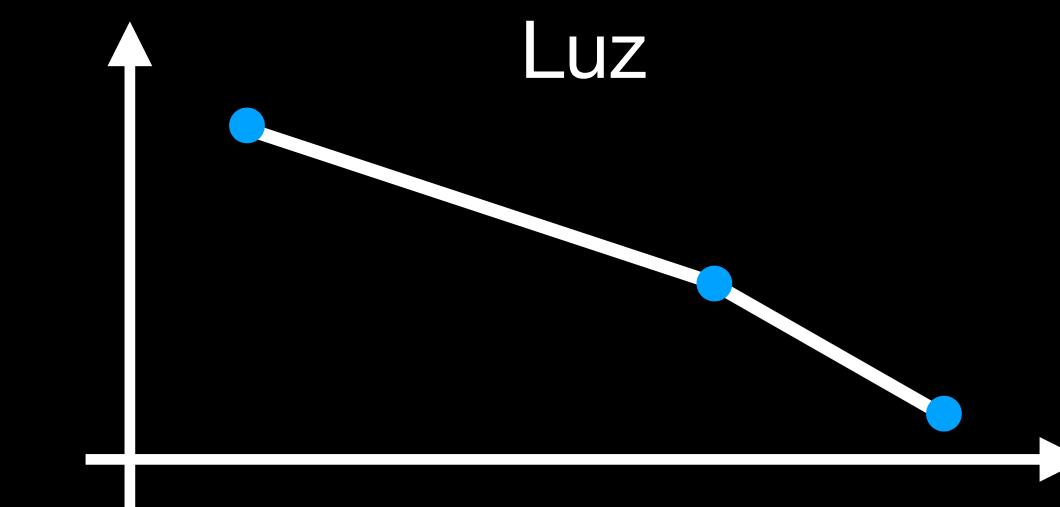
# Timescale

TimescaleDB

Data	Luz	Movimento
20/09/2024 16:30	74.5	1
20/09/2024 16:31	76.8	0
20/09/2024 16:32	80.3	0
20/09/2024 18:13	60.4	1
20/09/2024 18:33	57.4	1
20/09/2024 19:30	50.2	0



Data	Média Luz	Soma Movimento
de 16:00 às 16:59	77.2	1
de 17:00 às 17:59	NULL	NULL
de 18:00 às 18:59	58.9	2
de 19:00 às 19:59	50.2	0



Necessidade de Buscas com Janelamento em Séries Temporais

A screenshot of a web browser displaying the GitHub repository for TimescaleDB. The URL in the address bar is `github.com/timescale/timescaledb`. The page features a large logo icon on the left, followed by the word "Timescale" in a large, bold, black sans-serif font. Below the title, there is a descriptive text block: "TimescaleDB is a PostgreSQL extension for high-performance real-time analytics on time-series and event data". At the bottom of this section are three buttons: "READ THE TIMESCALE DOCS", "ASK THE TIMESCALE COMMUNITY", and "TRY TIMESCALE FOR FREE". The "TRY TIMESCALE FOR FREE" button includes a small icon of a computer monitor.

**Install TimescaleDB**

---

Install from a Docker container:

1. Run the TimescaleDB container:

TimescaleDB: Plugin no PostgreSQL Otimizado para Séries Temporais

A data\_hora tem que estar na primary key das hipertabelas.



```
CREATE TABLE dados (
    data_hora TIMESTAMP WITH TIME ZONE DEFAULT now(),
    luz FLOAT,
    movimento INT,
    PRIMARY KEY (data_hora)
);
```

+

```
SELECT create_hypertable('dados', 'data_hora');
```

```
SELECT data_hora, luz, movimento  
FROM dados  
WHERE data_hora > NOW() - INTERVAL '1 day'  
ORDER BY data_hora ASC;
```



```
SELECT  
    time_bucket('1 hour', data_hora) AS time,  
    AVG(luz) AS media_luz,  
    SUM(movimento) AS soma_movimento  
FROM dados  
WHERE data_hora > NOW() - INTERVAL '1 day'  
GROUP BY time  
ORDER BY time ASC;
```

Comando SQL mais Específico para Janelamento em Hipertabelas

The screenshot shows a web browser window displaying the TimescaleDB API documentation at [docs.timescale.com/api/latest/](https://docs.timescale.com/api/latest/). A search bar at the top contains the query "first". Below the search bar, there are 12 results found. The results are categorized into two sections: "Hyperfunctions: percentile approximation: uddsketch()" and "Hyperfunctions: general".

**Hyperfunctions: percentile approximation: uddsketch()**

- `approx_percentile`: Estimate the value at a given percentile from a `uddsketch`.
- `approx_percentile_array`: Estimate the values for an array of given percentiles from a `uddsketch`.
- `approx_percentile_rank`: Estimate the percentile of a given value from a `uddsketch`.
- `error`: Get the maximum relative error for a `uddsketch`.
- `mean`: Calculate the exact mean from values in a `uddsketch`.
- `num_vals`: Get the number of values contained in a `uddsketch`.
- `percentile_agg`: Aggregate data in a `uddsketch`, using some reasonable default values, for further calculation of percentile estimates.
- `rollup`: Roll up multiple `uddsketch`s.
- `uddsketch`: Aggregate data in a `uddsketch` for further calculation of percentile estimates.

**Hyperfunctions: general**

- `approximate_row_count`: Estimate the number of rows in a table.
- `days_in_month`: Calculates days in month given a timestamp.
- `first`: Get the first value in one column when rows are ordered by another column.
- `histogram`: Partition the dataset into buckets and get the number of counts in each bucket.
- `last`: Get the last value in one column when rows are ordered by another column.

On the right side of the search results, there is a "TimescaleDB API reference" link and a "Was this page helpful?" section with "Yes" and "No" buttons. The sidebar on the left includes links for "Get started", "About Timescale products", "Use Timescale", "Tutorials", "Integrations", and "API Reference".

## Outros Comandos do TimescaleDB



Grafana



# Diversos Tipos de Visualização de Dados

Home

Starred

Dashboards

Explore

Alerting

Connections

Add new connection

Data sources

Administration

## Add data source

Choose a data source type

sql

 Microsoft SQL Server  
Data source for Microsoft SQL Server compatible databases  
Core

 MySQL  
Data source for MySQL databases  
Core

 PostgreSQL  
Data source for PostgreSQL and compatible databases  
Core

**Connection**

Host URL \*  
localhost:5432

Database name \*  
Database

**Authentication**

Username \*  
Username

Password \*  
Password

TLS/SSL Mode ⓘ  
require

TLS/SSL Method ⓘ  
File system path

**Additional settings**

PostgreSQL Options

Version ⓘ  
15

Min time interval ⓘ  
1m

TimescaleDB ⓘ

Passo 1: Escolha da Fonte de Dados

The image shows the Grafana interface. On the left, a sidebar menu includes Home, Starred, Dashboards (selected), Explore, Alerting, and Connections. The main area is titled 'Dashboards' with the subtitle 'Create and manage dashboards to visualize your data'. It features a search bar, a 'Filter by tag' dropdown, a 'Starred' checkbox, and sorting options. A large central box contains the text 'Start your new dashboard by adding a visualization' and a 'Add visualization' button. A modal window titled 'Select data source' is open, showing a search bar with 'sql' typed in. Two data sources are listed: 'Janks PostgreSQL' and 'grafana-postgresql-datasource'. A hand cursor icon is positioned over the 'grafana-postgresql-datasource' entry. A second hand cursor icon is positioned over the 'New ^' button in the top right corner of the main dashboard area.

Home

Starred

Dashboards

Explore

Alerting

Connections

## Dashboards

Create and manage dashboards to visualize your data

Search for dashboards and folders

Filter by tag

Starred

New ^

New dashboard

New folder

Import

Start your new dashboard by adding a visualization

Add visualization

Select data source

sql

Janks PostgreSQL

grafana-postgresql-datasource

Passo 2: Criação de um Dashboard

Panel Title

No data

Query 1 Transform data 0 Alert 0

Data source Janks PostgreSQL

MD = auto = 685 Interval = 30s

Query inspector

Run query

escreva a query SQL aqui!

Time series

Search options

All Overrides

Panel options

Title

Description

Transparent background

Panel links

Repeat options

Tooltip

Legend

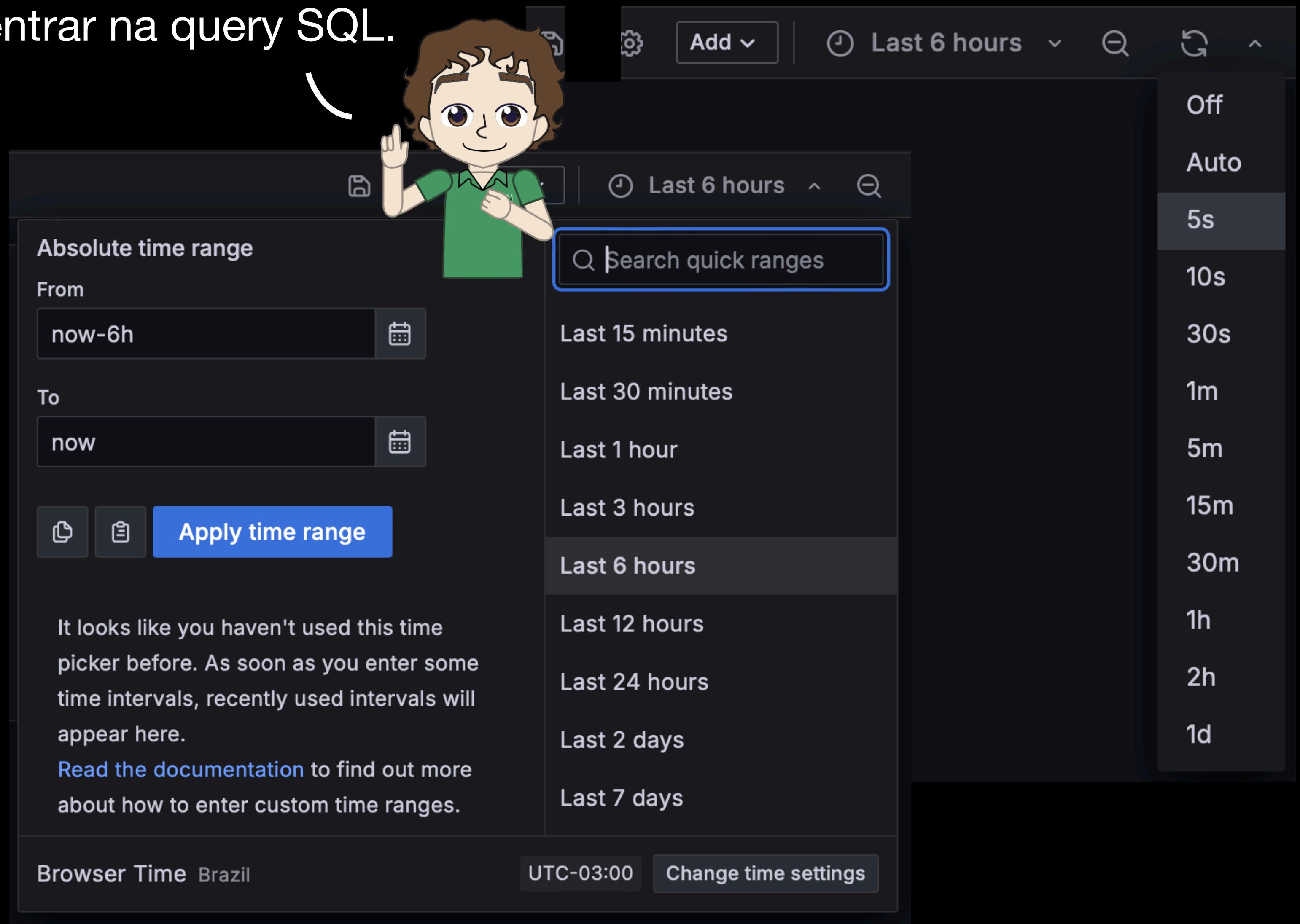
Axis

Time zone

Default

Passo 3: Configuração da Visualização dos Dados

A faixa de tempo escolhida  
tem que entrar na query SQL.



A screenshot of the Grafana interface focusing on the time range selection. A cartoon character with brown hair and a green shirt is pointing towards the dropdown menu. The top navigation bar shows 'Add ▾' and 'Last 6 hours'. Below it, a secondary time range selector also shows 'Last 6 hours'. A search bar labeled 'Search quick ranges' is present. A large dropdown menu lists various time intervals. The 'Last 6 hours' option is highlighted. Other options include 'Last 15 minutes', 'Last 30 minutes', 'Last 1 hour', 'Last 3 hours', 'Last 12 hours', 'Last 24 hours', 'Last 2 days', and 'Last 7 days'. To the right of the dropdown, a vertical column of time intervals is listed: Off, Auto, 5s, 10s, 30s, 1m, 5m, 15m, 30m, 1h, 2h, and 1d. At the bottom left, there's a note about using the time picker for the first time, and at the bottom right, there are buttons for 'Browser Time Brazil', 'UTC-03:00', and 'Change time settings'.

Absolute time range

From: now-6h

To: now

Apply time range

It looks like you haven't used this time picker before. As soon as you enter some time intervals, recently used intervals will appear here.

Read the documentation to find out more about how to enter custom time ranges.

Last 15 minutes

Last 30 minutes

Last 1 hour

Last 3 hours

Last 6 hours

Last 12 hours

Last 24 hours

Last 2 days

Last 7 days

UTC-03:00

Change time settings

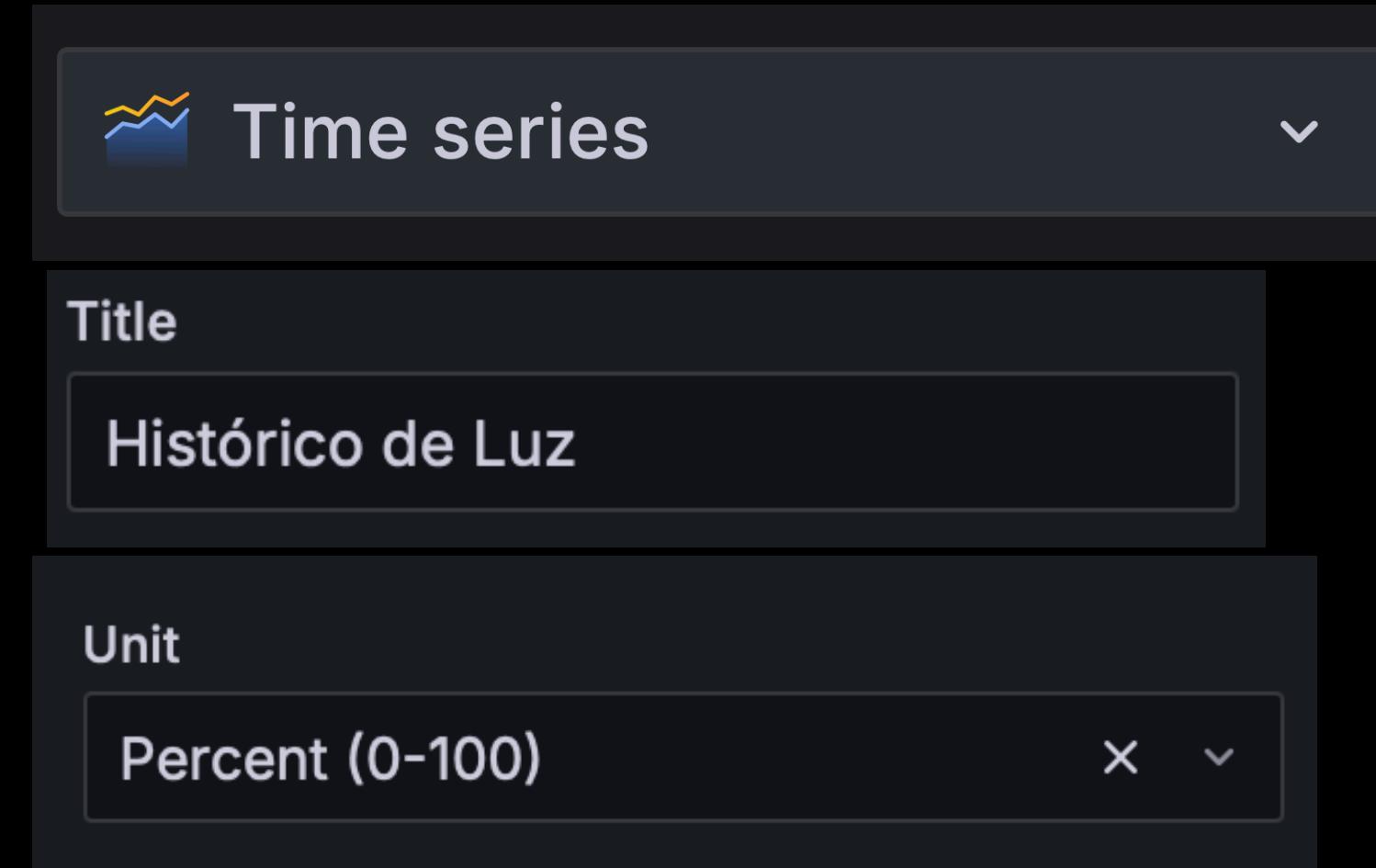
Menu de Faixa de Tempo e de Tempo de Atualização Automática

```
SELECT
    time_bucket('1 hour', data_hora) AS time,
    AVG(luz) AS media_luz,
FROM dados
WHERE data_hora > NOW() - INTERVAL '1 day'
GROUP BY time
ORDER BY time ASC;
```



```
SELECT
    time_bucket('1 hour', data_hora) AS time,
    AVG(luz) AS media_luz
FROM dados
WHERE $__timeFilter(data_hora)
GROUP BY time
ORDER BY time ASC;
```

Exemplo de Query para Série Temporal em Gráfico de Linha



A (PostgreSQL - Supermercado)

Format: Table ▾

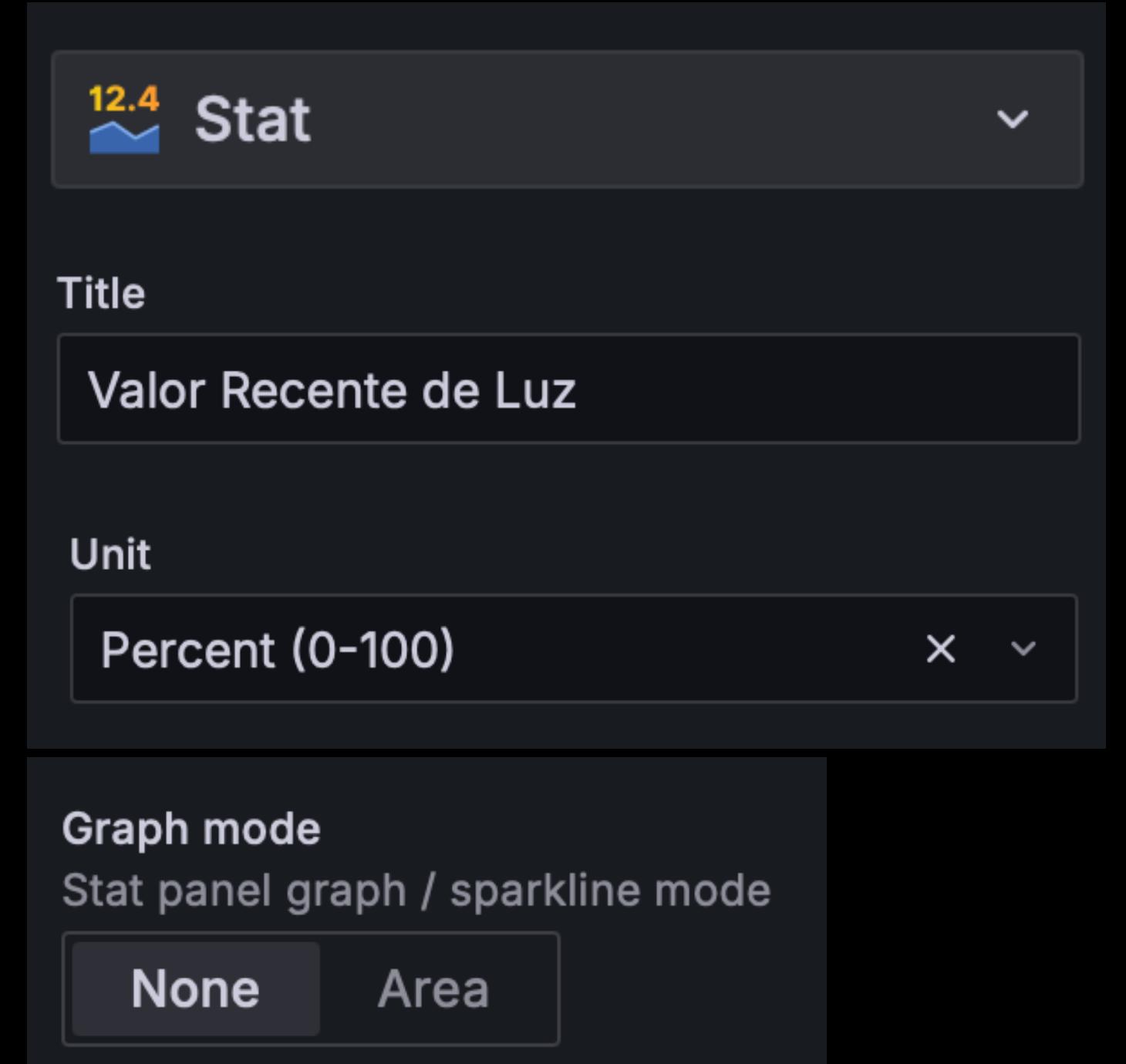
Run query Builder Code

```

1 SELECT
2     time_bucket('1 hour', data_hora) AS time,
3     AVG(luz) AS media_luz
4 FROM dados
5 WHERE $__timeFilter(data_hora)
6 GROUP BY time
7 ORDER BY time ASC;
8
9
10
11
12
13

```

Exemplo de Gráfico de Linha para Série Temporal

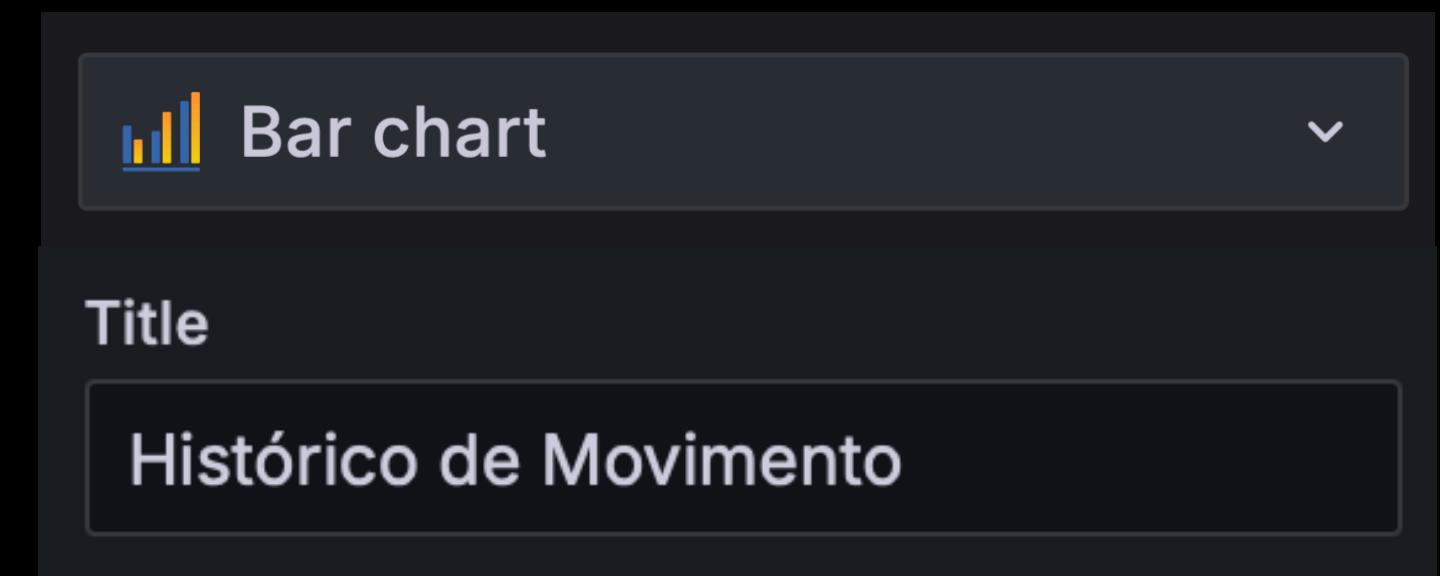
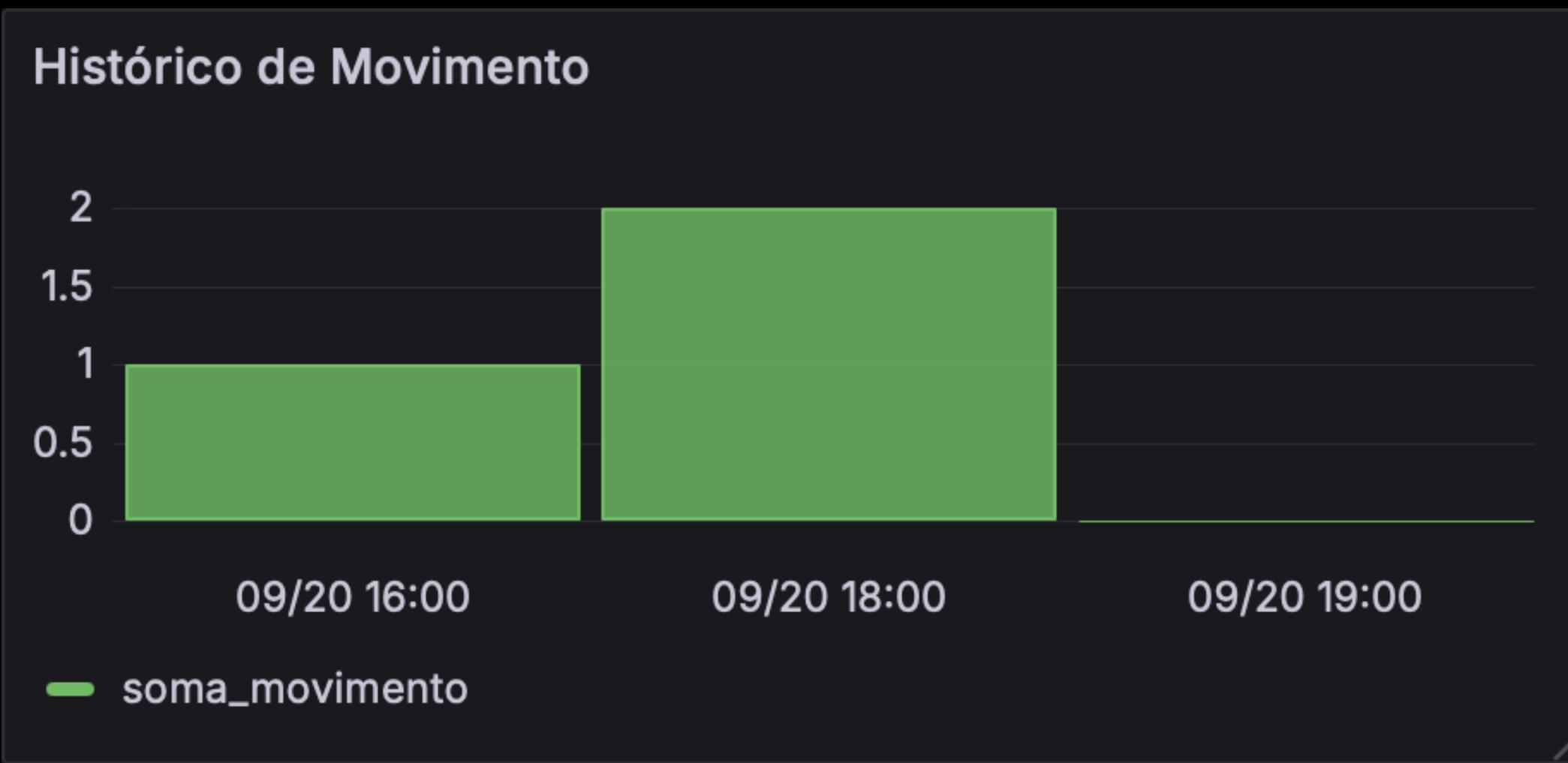


A (PostgreSQL - Supermercado)

Format: Table Run query Builder Code

```
1 SELECT
2     time_bucket('1 hour', data_hora) AS time,
3     AVG(luz) AS media_luz
4 FROM dados
5 WHERE $__timeFilter(data_hora)
6 GROUP BY time
7 ORDER BY time ASC;
```

Exemplo de Gráfico de Status



A (PostgreSQL - Supermercado)

Format: Table ▾

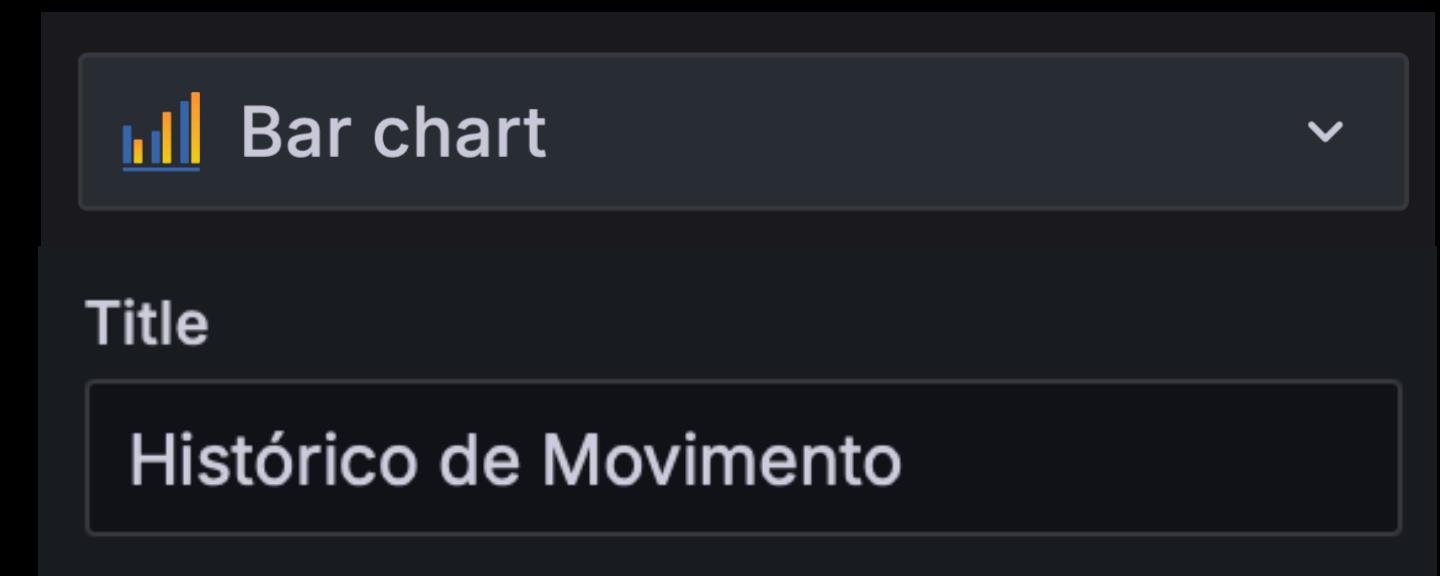
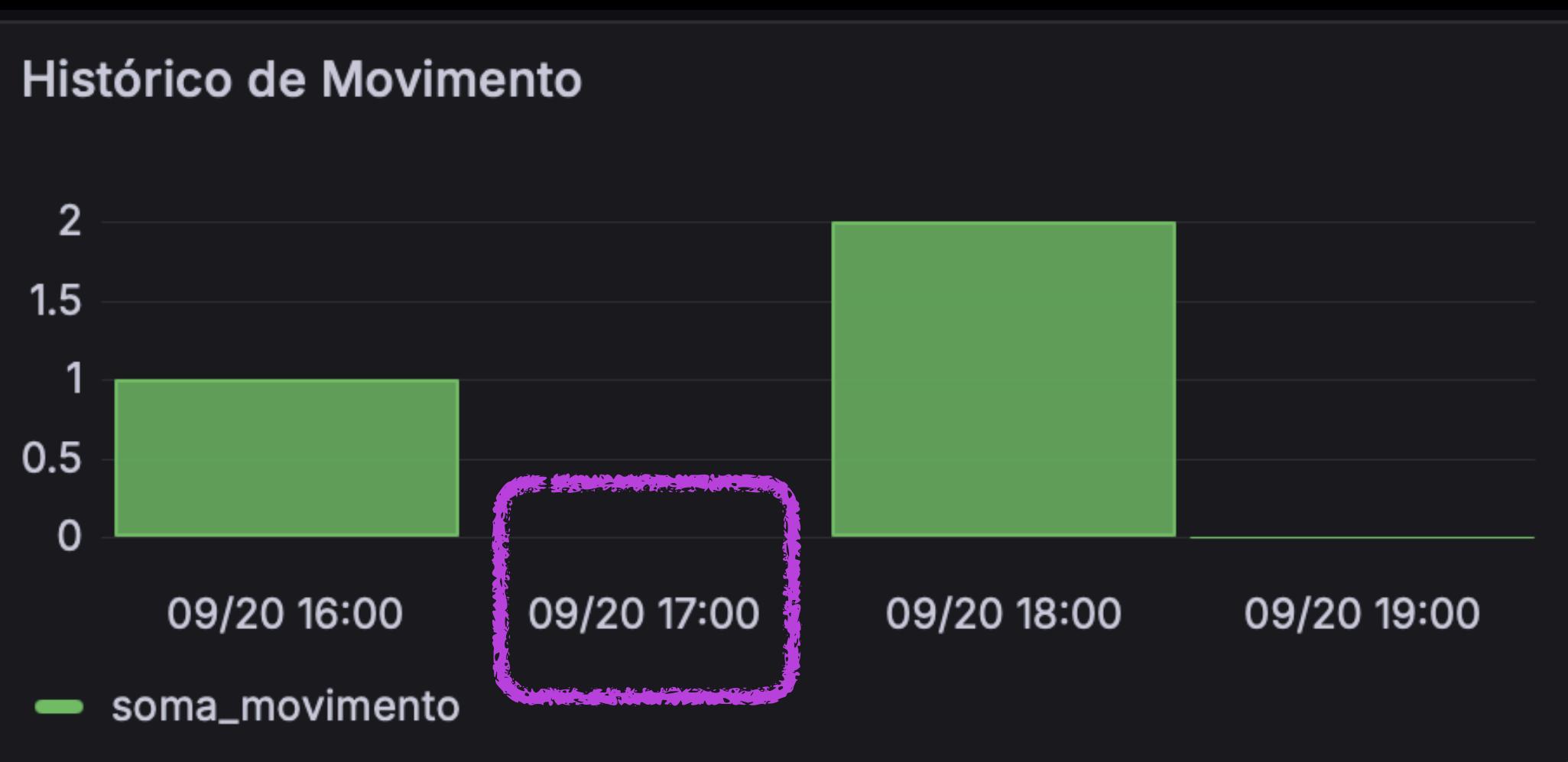
Run query Builder Code

```

1 SELECT
2   time_bucket('1 hour', data_hora) AS time,
3   SUM(movimento) AS soma_movimento
4
5 FROM dados
6
7 WHERE $__timeFilter(data_hora)
8
9 GROUP BY time
10
11 ORDER BY time ASC;
12
13

```

Exemplo de Gráfico de Barras



A (PostgreSQL - Supermercado)

Format: Table ▾

Run query Builder Code

```

1 SELECT
2   time_bucket_gapfill('1 hour', data_hora) AS time,
3   SUM(movimento) AS soma_movimento
4
5 FROM dados
6
7 WHERE $__timeFilter(data_hora)
8
9 GROUP BY time
10
11 ORDER BY time ASC;
12
13

```

Exemplo de Gráfico de Barras com Gapfill

# Settings

General

Annotations

Variables

Links

JSON Model



There are no variables added yet

Variables enable more interactive and dynamic dashboards. Instead of hard-coding things like server or sensor names in your metric queries you can use variables in their place. Variables are shown as list boxes at the top of the dashboard. These drop-down lists make it easy to change the data being displayed in your dashboard.

Check out the [Templates and variables documentation](#) for more information.

 Add variable

Variáveis para Outros Filtros Interativos

```
SELECT saldo  
FROM clientes  
WHERE nome = '$nomeCliente'
```

Nome do Cliente  Enter variable value

None

```
SELECT SUM(valor)  
FROM vendas  
WHERE tipo = '$tipoProduto'
```

Tipo de Produto  Enter variable value

Alimento  
Limpeza  
Cosméticos  
Roupas

Select variable type

General

Name  The name of the template variable. (Max. 50)

Label  Optional display name

Select variable type

General

Name  The name of the template variable. (Max. 50)

Label  Optional display name

Custom options  
Values separated by comma

Selection options

Multi-value  
Enables multiple values to be selected at the same time.

Include All option  
Enables an option to include all variables

Configuração das Variáveis para as Buscas SQL