



## Desafio Técnico Front- end

## Desafio Técnico – Front-End

Olá! Agradecemos pelo seu interesse em fazer parte da equipe da MOB4AI, atuando na área de desenvolvimento e pesquisa. A seguir, você encontrará todas as informações necessárias para a realização do desafio técnico.

---

### Instruções Gerais

- Leia atentamente todas as orientações antes de iniciar o desafio;
  - Siga o máximo possível as instruções descritas neste documento;
  - Crie um repositório público no seu GitHub pessoal para hospedar o projeto MOB4AI;
  - É proibido o uso intensivo de IA;
  - Realize os commits ao longo do desenvolvimento no mesmo repositório;
  - É permitido consultar fontes externas como Stack Overflow, documentações oficiais, Google ou projetos próprios para auxiliar na implementação.
- 

### Descrição do Projeto

Seu desafio será construir um **dashboard interativo** para análise de dados relacionados à **bateria e temperatura**. A api do projeto roda um backend em Ktor via Docker para fornecer dados de bateria e temperatura.

Como executar:

1. Clone o repositório: <https://github.com/vmelchiors/DesafioFrontend.git>
2. `docker-compose up --build`
3. Acesse no navegador

O backend possui as seguintes rotas:

1. `http://localhost:8080/` → Página de hello world
2. `http://localhost:8080/temperature` → Dados de temperatura
3. `http://localhost:8080/battery` → Dados de bateria

Os dados serão fornecidos no seguinte formato:

- ❖ Rota `/battery`

```
{
  "timestamp": 1621549200000,
  "rem_cap": 50,
  "inst_curr": -287,
  "voltage": 3773,
  "battery_level": 80,
  "battery_status": 2,
  "plug_type": 1
}
```

Descrição dos campos:

1. timestamp: Marca temporal em milissegundos.
2. rem\_cap: Capacidade restante da bateria, em percentagem.
3. inst\_curr: Corrente instantânea em miliampère-hora (mAh).
4. voltage: Tensão da bateria em milivolts (mV).
5. battery\_level: Nível da bateria em percentagem.
6. battery\_status: Status da bateria.

❖ Rota /temperature

```
{
  "timestamp": 1621549200000,
  "temp_bat": 27,
  "temp_front": 25,
  "temp_back": 26,
  "temp_cpu": 45
}
```

Descrição dos campos:

1. timestamp: Marca temporal em milissegundos.
2. temp\_bat: Temperatura da bateria em graus Celsius.
3. temp\_front: Temperatura da frente do dispositivo.
4. temp\_back: Temperatura da parte de trás do dispositivo.
5. temp\_cpu: Temperatura da CPU em graus Celsius.

---

## Requisitos

### 1. Gráficos (Dashboard)

O dashboard deve conter **quatro gráficos**, cada um representando um dos indicadores abaixo:

- **Corrente Instantânea:** Gráfico de linha
- **Nível de Bateria:** Gráfico de barras ou de área
- **Temperatura da Bateria:** Gráfico de linha

- **Temperatura da CPU:** Gráfico de linha

Cada gráfico deve exibir uma **tooltip ao passar o mouse**, com as seguintes informações:

- **Data:** DD/MM/AA - HH:MM:SS
  - **Corrente Instantânea (Instant Current):** X mAh
  - **Capacidade (Capacity):** X %
  - **Temperatura da Bateria (Battery Temperature):** X °C
  - **Temperatura da CPU (CPU Temperature):** X °C
- 

## 2. Barra Lateral

O projeto deve conter uma **barra lateral sincronizada com os gráficos ao passar o mouse**, com dados estruturados a partir das informações fornecidas no JSON. Abaixo, segue um exemplo da estrutura esperada:

Plug Type/ Status	AC	Charging
Voltage	3773	Mv
Instant Current	-287	mAh
Temperature Bat	27	°C

## 3.Formatção de Dados

Alguns campos do JSON devem ser **convertidos para texto legível**, conforme as tabelas abaixo. Essas informações devem ser apresentadas de forma clara na **barra lateral** ou onde for mais adequado no layout.

### Plug Type (Tipo de Conexão)

Valor	Descrição
0	Unplugged
1	AC
2	USB
3	Wireless

### Battery Status (Status da Bateria)

Valor	Descrição
1	Unknown
2	Charging
3	Discharging
4	Not Charging

5	Full
6	Wireless

- Importante: essas conversões devem ser feitas antes de exibir os dados no dashboard. Por exemplo, se PlugType for 2, o texto exibido deve ser "USB".