

é preciso alinhar as funções fetch do javascript que declaram os métodos post e get com duas funções diferentes no backend.

A função post atualmente é composta por uma gambiarra do gemini que na hora pareceu uma boa ideia:

```
'description': 'Receives a barcode, consults Open Food Facts, calculates score, and saves.'
```

Isso significa que dentro da função post existe a chamada GET para a API externa. Isso está gerando um conflito com a função fetch do javascript, que não consegue escolher qual método realizar.

Desse modo, preciso que a função POST seja completamente separada da função GET pra api externa

Isto significa que a primeira função vai ter que ser uma get para a OFF, diferente da GET interna, o que não impede de haver uma condicional para checar se o produto já existe na base de dados e, se sim, rodar a função get interna.

RASCUNHO

```
@product_bp.route('/product', ['GET'])
def get_product_db(barcode):
    if product in db:
        response = request.get_json()
    else:
        scan_product_off()

@product_bp.route('/product', methods=['GET'])
def scan_product_off(barcode):
    if product in db:
        get_product_db()
    else:
        API_URL = ...
        response =
        data = request.get_json().json
    return data

@product_bp.route('/product', methods=['POST'])
def post_product_db(data):
    post data in db
```

ALGORITMO product_bp:

- escaneia codigo de barras (barcode)
- verifica (if) se ja esta na base de dados - funcao GET com o barcode
- se (if) o produto ja estiver, faz uma requisicao GET ao banco de dados
- (ELSE) se o produto nao estiver na base de dados, faz a requisicao GET pra api externa OFF e retorna os dados em json (data)
- calcula o score baseado nos dados da json -> calculate_score()
- pega os dados em json e o score e cria um novo produto na base de dados com uma funcao POST