

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

a funcao 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 funcao post existe a chamada GET para a API externa. Isso esta gerando um conflito com a funcao fetch do javascript, que nao consegue escolher qual metodo realizar.

desse modo, preciso que a funcao POST seja completamente separada da funcao GET pra api externa

isso significa que a primeira funcao vai ter q ser uma get para a OFF, diferente da GET interno, o q nao impede de haver uma condicional para checar se o produto ja existe na base de dados e, se sim, rodar a funcao 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 código de barras (barcode)
- verifica (if) se já está na base de dados - função GET com o barcode
- se (if) o produto já estiver, faz uma requisição GET ao banco de dados
- (ELSE) se o produto não estiver na base de dados, faz a requisição GET para a 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 função POST