

Atividade 1: Planejamento e Estrutura do Microserviço

Endpoint POST para cadastrar um consumo, retornando o código correto para cada caso, por exemplo para dados de consumo invalido, consumo criado e quando o ocorrer um erro também, mesma coisa para o endpoint GET de consumo e um endpoint para conferir a saúde, retornando simplesmente que o servidor está no ar.

```
[HttpPost("consumo")]
0 referências
public async Task<IActionResult> RegistrarConsumo([FromBody] ConsumoRequest request)
{
    if (request == null || request.Quantidade <= 0)
        return BadRequest("Dados de consumo inválidos.");

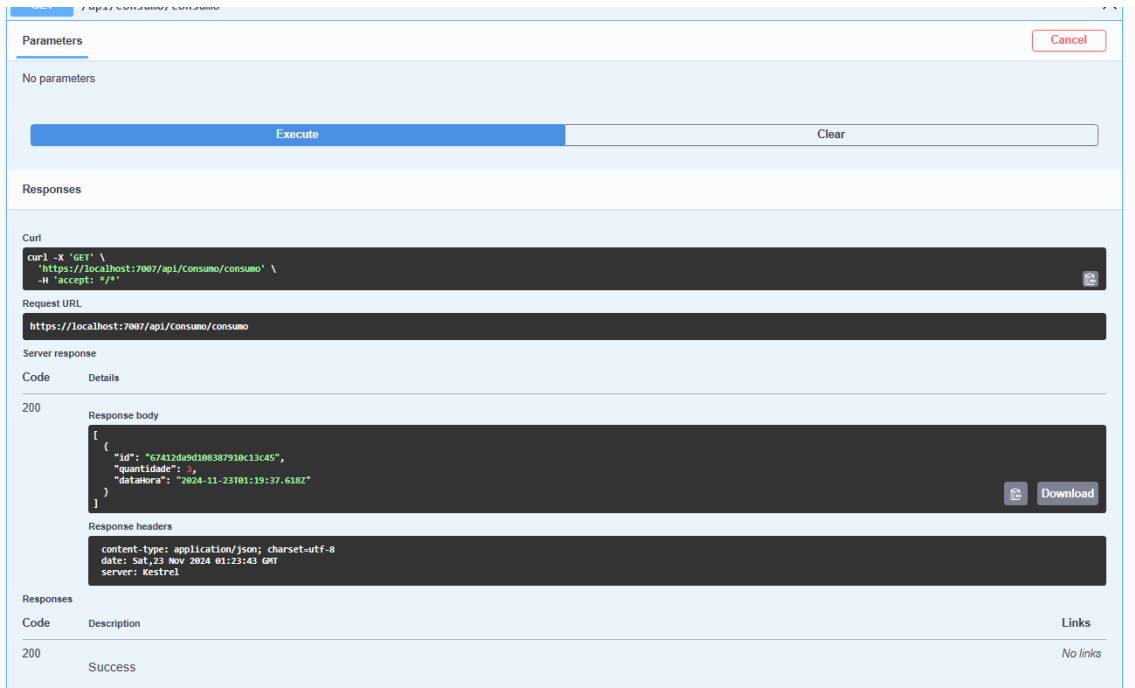
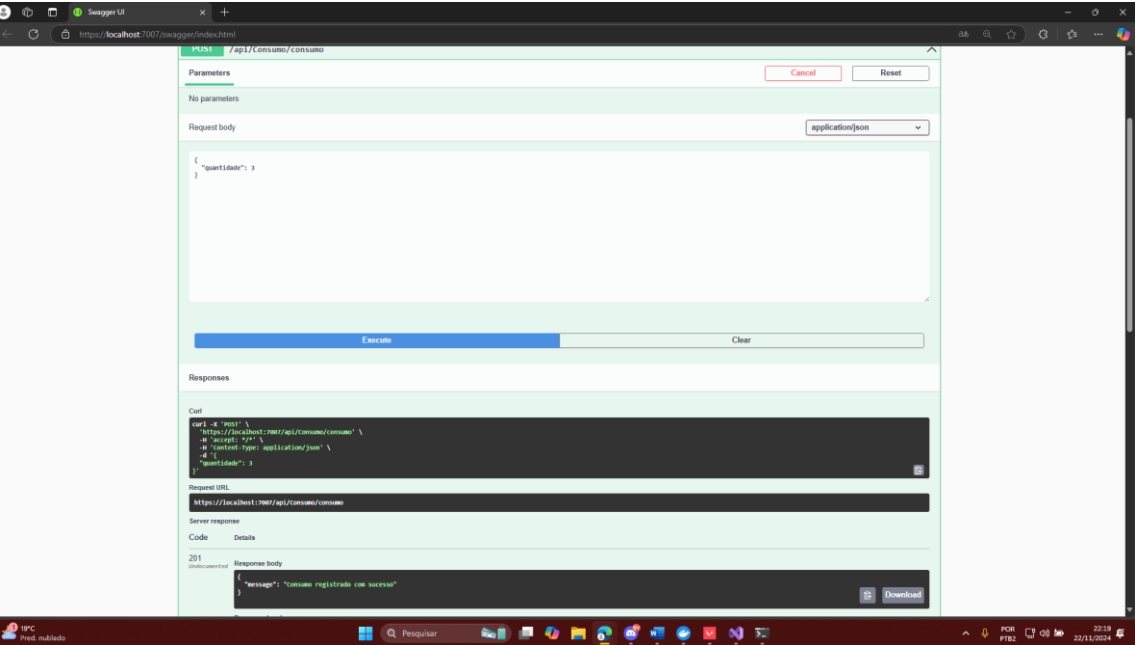
    try
    {
        await _consumoService.RegistrarConsumoAsync(request);
        return Created("", new { message = "Consumo registrado com sucesso" });
    }
    catch (Exception ex)
    {
        return StatusCode(500, new { error = ex.Message });
    }
}
```

```
[HttpGet("consumo")]
0 referências
public async Task<IActionResult> ConsultarConsumo()
{
    try
    {
        var consumos = await _consumoService.ObterConsumosAsync();
        if (consumos == null || !consumos.Any())
            return NotFound("Nenhum consumo encontrado.");

        return Ok(consumos);
    }
    catch (Exception ex)
    {
        return StatusCode(500, new { error = ex.Message });
    }
}
```

Atividade 2: Integração com MongoDB

O print abaixo demonstra o cadastro de um registro sendo realizado com sucesso e o código demonstra que existe a conexão com o banco de dados, depois na segunda print, podemos ver o registro cadastrado sendo retornado.



2 referências

```
public async Task RegistrarConsumoAsync(ConsumoRequest request)
{
    var consumo = new ConsumoEnergetico
    {
        Id = ObjectId.GenerateNewId().ToString(),
        Quantidade = request.Quantidade,
        DataHora = DateTime.UtcNow
    };

    await _repository.SalvarAsync(consumo);

    // Remove o cache após inserção de novos dados
    await _cacheService.RemoveAsync(CacheKey);
}
```

Abaixo utilização do cache

Atividade 3: Implementação de Cache com Redis

Primeira print mostra o código sendo executado no postman e a diferença de velocidade entre sem a utilização do cache e com o cache.

