



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
DIRETORIA DE EDUCAÇÃO CONTINUADA

Curso/Oferta: Arquitetura de Software Distribuído – Oferta 11

Disciplina: Arquitetura de Backend e Microsserviços

Professor: Marco Mendes

Data: 12/04/19

Nomes: Juan Requeijo Silva Carvalho

Atividade Aula 02

- **Postar comentários para livros. (resenhadas)**
- **Pesquisa de livros por critérios diversos.**
- **Manipular um carrinho de compras**
- **Realizar pedidos**
- **Acompanhamento o status das entregas realizadas.**
- **Cadastrar livros**

Desenho da sua API

Cadastro de livros

api/livros/

Métodos: POST, GET

api/users

Métodos: POST, GET

Acompanhamento o status das entregas realizadas.

api/status

Métodos: GET

Realizar pedidos

api/demands/

Métodos: POST

Manipular um carrinho de compras

api/checkout

Métodos: PUT, GET, POST, DELETE

Pesquisa de livros por critérios diversos

api/search

Método: GET.

Postar comentários para livros.

api/comments

Método: GET, POST

Código fonte (Java ou .NET) da API da gestão de livros publicado no GitHub.

<https://github.com/juanrequieijo/library.git>

Justificativas que as 11 regras apresentadas no capítulo 3 foram observadas.

- 1- Organize APIs ao longo de recursos – A api foi organizada para ser um espelho das tabelas do banco de dados.
- 2- Padronize as suas APIs. – A api foi padronizada para ter poucos níveis, e assim ficar mais simplificada.
- 3- Evite APIs anêmicas – essa arquitetura é anêmica pois expõe a estrutura física da API, e a reduz a um CRUD.
- 4- Crie APIs simples – Estrutura simples baseadas nas tabelas.
- 5- Considere a atualização em lote para operações complexas – Edição de carrinho por PUT.
- 6- Se você precisar receber datas e horas nas API, use o padrão ISO 8601 – esta api não precisa receber datas.
- 7- Documente sua API. – Swagger foi implementada
- 8- Use protocolo HTTPS/SSL. – VS code por padrão habilita o https.
- 9- Versione suas APIs – Controle de versionamento no GIT, não foi adicionada na URL a versão.

- 10- Estabeleça paginação para coleções com grandes volumes de dados – Não foi adicionada a paginação.
- 11- Use corretamente os códigos de retorno HTTP – Os códigos de status HTTP 404 para não encontrado um GET, 200, 201 de inserção com sucesso por POST, 204 por edição por PUT, 204 com DELETE com sucesso.