

FIAP

Arthur Coutinho Santos

RM 336256

Persistence - Avaliação

São Paulo

2020

SUMÁRIO

| | |
|-----------------------|---|
| 1. Repositórios | 3 |
| 2. Ferramentas: | 3 |
| 3. Prints Back | 4 |
| 4. Modelo do database | 5 |

1. Repositórios

<https://github.com/ArtCouSan/Projeto-Persistence>

2. Ferramentas:

- Back-end Java e Spring Boot (Spring Data + Redis)

Foi utilizado o Redis pela facilidade de implementação no projeto e sua fácil integração, além de se encaixar bem nos requisitos apresentados, é claro não descartando a hipótese de usar outros métodos de armazenamento de cache.

Apliquei nas Entidades de Cliente e Produto:

1. Cliente para caso não seja implementado sessão
2. Produto para evitar buscas e listagem diretas na base, porém sempre tomando o cuidado de apagar o cache caso seja alterado as informações, como no caso da realização de um pedido, onde é apagado os registros do Redis

As **informações da aplicação podem ser facilmente alteradas pelo application.yaml**, exemplo porta do Redis ou MySQL.

- MySQL como database

Deve ser criado no seu MySQL o banco com o nome ecommerce ou o que foi definido no application.yaml

- Swagger como documentação da API
- Versionamento no Github
- Maven gerencia dependências

3. Prints Back

| | | |
|--|------------------|----------------------|
| ordered-controller Ordered Controller | | ▼ |
| POST | /v1/order | Make order |
| product-controller Product Controller | | ▼ |
| GET | /v1/product | List all products |
| POST | /v1/product | Save product |
| GET | /v1/product/{id} | Find product by id |
| PUT | /v1/product/{id} | Update product by id |
| DELETE | /v1/product/{id} | Delete product by id |
| users-controller Users Controller | | ▼ |
| GET | /v1/user | List all users |
| POST | /v1/user | Save user |
| GET | /v1/user/{id} | Find user by id |
| PUT | /v1/user/{id} | Update user by id |
| DELETE | /v1/user/{id} | Delete user by id |

4. Modelo do database

