

Projeto Final - Luiza<code>

Carrinho de compras

Visão Geral

No projeto final, cada grupo irá criar uma aplicação servidora de APIs REST. A aplicação irá *fornecer* um conjunto de APIs para um carrinho de compras de uma determinada categoria.

Entendam “categoria” como sendo um *negócio* de venda; por exemplo, quem for trabalhar com a categoria “Músicas”, irá criar para o seu projeto um carrinho de compras que “venda músicas”.

Vocês irão personalizar o seu trabalho conforme a sua categoria, sem a necessidade de criar uma aplicação de frente (front end) para este projeto final.


A seguir iremos apresentar mais detalhes do projeto final, informando o que é **obrigatório** e o que é *opcional* (mas que pode ser um diferencial) na composição do projeto final.

Objetivo Geral

Criar um projeto Python que seja um conjunto de APIs REst para um carrinho de compras, utilizando o *framework* FastAPI e que tenha os seus registros salvos no banco de dados MongoDB.

As informações que teremos no projetos são:

- Clientes: um cliente é a pessoa que está realizando a compra de um ou mais produtos no nosso sistema.
- Produtos: Produto que o cliente deseja adquirir. De um trabalho para outro poderá variar em seus detalhes. Por exemplo, um livro tem ISBN, enquanto que um esmalte possui uma cor.
- Carrinho de compras: *Informação central* do projeto, em que um cliente *agrupa* um mais produtos.



Seu projeto precisará criar, atualizar e até remover estas informações.

Requisitos Funcionais


Cadastro de clientes

Permitir o gerenciamento de clientes, para isso **deve** ser possível:

1. Cadastrar um cliente.
 - a. Processo em que inserimos um novo cliente no sistema.
 - b. Cada cliente precisa ter pelo menos um nome e um e-mail.
 - c. O cliente deve informar um email válido (ao menos 3 caracteres, conter um @)
 - d. O e-mail do cliente deve ser único, ou seja, não há dois clientes no sistema com o mesmo e-mail.
 - e. Podemos ter dois clientes com o mesmo nome; mas, cada um com um e-mail diferente.
2. Cadastrar um endereço.
 - a. Processo de inserir um endereço para o cliente.
 - b. Cada endereço precisa ter pelo menos um CEP, logradouro, número, cidade e estado..
 - c. O mesmo endereço pode ser cadastrado para mais de um cliente.
3. Pesquisar um cliente.
 - a. Informado o e-mail de um cliente, apresentamos os seus dados.
4. Pesquisar um endereço.
 - a. Informando o e-mail de um cliente, apresentamos seus endereços.
5. Remover um cliente. (Opcional)
 - a. Remover um cliente pode remover também suas outras informações (ex: endereços).
 - b. Ao remover um cliente, atente-se ao seguinte: Este cliente possui carrinhos de compras? Se sim, o que faço? Você decide!
6. Remover um endereço. (Opcional)
 - a. Com base em alguma chave, por exemplo, é possível remover o endereço do cliente.

Gerenciamento de produtos

Conter o gerenciamento de produtos. Solicitamos o seguinte:

- 
1. Cadastrar um produto.
 - a. Processo em que registra-se um novo produto no sistema.
 - b. Cada produto precisa ter pelo menos um nome, uma descrição e um código único.
 - c. Um produto pode ter um preço de venda, que é um valor superior a R\$ 0,01.
 - d. Um produto pode ter um valor de estoque, que é um valor superior a 0.
 - e. O código do produto informado no processo de cadastro deve ser único, ou seja, não há dois produtos no sistema com o mesmo código.
 - f. Os nomes dos produtos podem ser únicos.
 - g. Campos adicionais podem ser informados conforme a **categoria** do seu projeto.
 2. Atualizar os dados de um produto.
 - a. Poderemos atualizar os dados de um produto com base no seu código.
 - b. O código do produto não pode ser alterado.
 - c. O nome do produto pode ser alterado.
 - d. Quaisquer outros campos do produto que existam em seu projeto poderão ser atualizados.
 3. Pesquisar um produto.
 - a. Informado o código do produto, apresentamos os seus dados.
 4. Pesquisar um produto pelo nome.
 - a. Informe um texto para o nome do produto, então iremos pesquisar pelos produtos que contenham o nome informado.
 5. Remover um produto. (Opcional)
 - a. Poderemos remover um produto com base em seu código.
 - b. Atentar neste caso: Este produto está em algum carrinho de compras? O que fazer? O seu grupo irá decidir.

Carrinho de compras

Conter o gerenciamento de um carrinho de compras, desde a sua criação até a geração de um pedido. Podemos ter dois tipos de carrinhos de compra em nosso sistema:

1. Abertos: Cada cliente pode ter apenas um carrinho de compras aberto na aplicação.
2. Fechados: São os carrinhos de compras os quais a compra já foi fechada.



Solicitamos o seguinte:

1. Criar um carrinho de compras **aberto** e adicionar itens ao carrinho.
 - a. Todo carrinho de compras deve conter um cliente.
 - b. É opcional, ter um produto inicialmente.
 - c. Se há um produto um mais produtos, na criação do carrinho, informe a quantidade de cada produto. No seu trabalho, você pode começar com apenas um produto.
 - d. Ao criar o carrinho, você deve:
 - i. Validar se o cliente existe
 - ii. Validar se o produto a ser adicionado no carrinho existe
 - iii. Verificar se o cliente já possui um carrinho aberto. Caso contrário criar um carrinho novo.
 - iv. Validar se a quantidade de itens do produto a ser adicionado no carrinho está disponível no estoque (opcional).
 - e. Ao adicionar um item no carrinho, o mesmo terá o valor total e quantidade de itens atualizado
2. Alterar a quantidade de itens do carrinho novo.
 - a. No carrinho novo, com base no produto informado, a quantidade é modificada.
 - b. Para isto, você irá:
 - i. Validar se produto existe no carrinho
 - ii. Validar se existe estoque para a quantidade desejada do produto (opcional).
 - iii. Atualizar o valor total e quantidade de itens do carrinho
 - c. Se o carrinho zerar o número de itens, ou seja, o cliente removeu todos os itens do carrinho, o mesmo pode ser excluído.
3. Consultar carrinho de compras aberto:..
 - a. Informar o cliente e retornar os dados do carrinho e produtos
4. Consultar os carrinhos fechados de um cliente (opcional).
5. Consultar os produtos e suas quantidades em carrinhos fechados (opcional).
6. Consultar quantos carrinhos fechados os clientes possuem (opcional).
7. Fechar o carrinho aberto:
 - a. Simplesmente pode-se mudar o *tipo* do carrinho de compras para “fechado”..

- b. Opcionalmente, o grupo pode adicionar o seguinte:
 - i. Identificar o endereço do cliente que será utilizado como o de entrega.
 - ii. Validar se o estoque pedido dos itens ainda está disponível, se estiver reduzir do estoque dos produtos a quantidade de itens do produto no carrinho (Muito opcional).
 - iii. Associar um *identificador* ao carrinho de compras como sendo o número do pedido.
- 8. Excluir carrinho do cliente (opcional).
 - a. Quer o carrinho seja aberto ou fechado, podemos remover o carrinho do sistema.

Entregas extras

Abaixo seguem algumas sugestões que podem ajudar a complementar o projeto de vocês. Estas serão mais um diferencial para a revisão e avaliação de seu projeto diante da banca avaliadora.. Além de poder destacar vocês para eventuais vagas no luizalabs ou outras empresas.

Sugestões para entregas extras::


- Especializar ou especificar bem o seu produto conforme a sua categoria.
- Documentar a API Rest com o Swagger/OpenAPI ou RedDoc.
- Organizar e estruturar o código do projeto.
- Montar um arquivo README.md detalhando o projeto com informações tais como:
 - Tecnologia utilizada.
 - Bibliotecas do Python que estão no projeto.
 - Membros da equipe.
 - Como montar o ambiente para executar a aplicação.
 - Detalhes do desenvolvimento do trabalho, informando do que foi solicitado o que foi feito (o que não foi feito) e os extras que foram feitos.
 - Informações extras sobre o trabalho.
- Criação de testes unitários.
 - No mercado de trabalho a cobertura de testes pode variar, um número bem utilizado pode ser de uma cobertura mínima de 80%.
- A API poderá utilizar algum recurso de autorização e autenticação.

- O grupo pode escolher o recurso desejado. Recomendamos o uso de **Basic** (cliente + senha) ou JWT.
- O projeto poderá gerar mensagens de log ou utilizar mecanismos de *logging*.
- Realizar o deploy da aplicação
 - A aplicação poderá ser hospedada em alguma plataforma como o Heroku e/ou ainda ter um arquivo Dockerfile ou docker-compose que ajude em seu futuro deploy/teste

Requisitos Técnicos

Seguem os requisitos técnicos para a realização do projeto:

- O grupo de trabalho deverá ser de no máximo X pessoas.
 - Os grupos serão sorteados.
- O código fonte do trabalho deverá estar em um repositório privado do Git.
 - O repositório deverá ser comum somente entre o grupo e depois entre os avaliadores do trabalho.
 - Será avaliado a participação de cada integrante do grupo (ou seja se as participantes realizaram commits em proporções parecidas)
- O projeto deverá ser feito utilizando Python.
- O módulo deverá ser uma API REST (com verbos HTTP e retorno (*status codes*) bem utilizados).
- Espera-se que o código esteja estruturado e organizado em arquivos, pastas e/ou funções (métodos e objetos). A estrutura ou organização poderá ser por:
 - Funcionalidade/Papel: Cada qual tem uma função (ou papel) específica para atender uma ou mais especificação.
 - Negócio: Temos uma estrutura separada para cada requisito funcional ou negócio. Por exemplo, uma pasta somente para os requisitos de produto.
 - Apesar de ser um requisito extra, este também é um requisito técnico, por ser um verdadeiro peso na revisão do projeto final.
- Deverá apresentar uma documentação a ser entregue juntamente com o código em que informa como instalar e configurar o ambiente para ser instalado em máquina local para realizar os testes necessários com a aplicação.

- 
- Poderá ser feita na documentação do README.md, da raiz do projeto. O README.md em si, é um requisito *opcional*, a documentação explicando para a banca como configurar e instalar o seu projeto, é obrigatória.
 - Data máxima para a entrega: **09/10/2022** às 23:59:59, horário de Brasília.
 - Avaliação
 - Datas: **10/10/2022** e **11/10/2022**.
 - A ordem e dia da apresentação serão sorteados.
 - O trabalho será apresentado para uma banca de avaliadores.
 - O grupo terá
 - 8 minutos para apresentar.
 - 5 minutos para responder às perguntas da banca.
 - 5 minutos para os comentários finais da banca.

Dicas

- Dividam entre vocês os itens obrigatórios, e deixem os itens opcionais como meta a ser atingida caso dê tempo ao fim do projeto.

