



LabECommerce

Enunciado

Seu nome anda circulando pelas redes e fóruns de desenvolvimento web! Seus projetos de front-end deram o que falar, e agora estão sabendo que você também apronta no back-end!

De manhã você acordou, tomou uma bela xícara de café, verificou sua caixa de e-mail e viu que a equipe do LabECommerce gostou muito do seu trabalho de front-end e quer te chamar para dar uma atualizada no back-end do sistema também.

Sendo assim, te passaram uma lista do que o projeto precisa ter:

Endpoints mínimos do MVP

Cadastro de usuário

A empresa quer implementar futuramente um sistema de login, por isso pediu que seja desenvolvido um cadastro simples de usuário nesse primeiro momento.

- **mysql:**
 - nome da tabela: `labecommerce_users`
 - colunas:
 - `id`: string (PRIMARY KEY)
 - `name`: string
 - `email`: string
 - `password`: string
- **express:**
 - método: `post`
 - path: `/users`
 - `id` deve ser gerado pela própria aplicação em código
 - parâmetros recebidos via `body`:
 - `name`, `email` e `password`

Busca por todos os usuários

Essa é uma funcionalidade de *admin* que será utilizada futuramente quando houver a classificação de papéis nas contas cadastradas (cliente comum e admin). Por enquanto, pediram apenas que o endpoint exista e funcione mesmo sem controle de segurança.

- método: `get`
- path: `/users`
- sem parâmetros
- deve trazer uma lista com **todos os usuários** cadastrados no banco

Cadastro de produto

Um dos principais requisitos do projeto. A equipe LabECommerce pediu que seja possível cadastrar produtos em um banco de dados e que o controle de identificadores seja gerenciado pelo próprio código.

- **mysql:**
 - nome da tabela: `labecommerce_products`
 - colunas:
 - `id`: string (PRIMARY KEY)
 - `name`: string

- `price` : number
- `image_url` : string
- express:
 - método: `post`
 - path: `/products`
 - `id` deve ser gerado pela própria aplicação em código
 - parâmetros recebidos via `body` :
 - `name` , `price` e `image_url`
 - `image_url` é a *url* da imagem do produto
 - exemplo de uma `image_url` :
 - <https://i.picsum.photos/id/0/5616/3744.jpg?hmac=3GAAioiQziMGtLbfrdbcoenXoWAW-zlyEAMkfEdBzQ>

Busca por todos os produtos

Essa é a funcionalidade principal de clientes. É o endpoint que irá possibilitar a exibição do catálogo de produtos cadastrados no website.

- método: `get`
- path: `/products`
- sem parâmetros
- deve trazer uma lista com **todos os produtos** cadastrados no banco

Registro de compra

É necessário para que seja possível controlar as vendas finalizadas. Por enquanto não se preocupe com confirmação de pagamento ou controle de datas. Uma implementação simples de registro já é suficiente para a equipe iniciar os testes de usabilidade.

- mysql:
 - nome da tabela: `labecommerce_purchases`
 - colunas:
 - `id` : string (PRIMARY KEY)
 - `user_id` : string (FOREIGN KEY) referencia uma `id` de `labecommerce_users`
 - `product_id` : string (FOREIGN KEY) referencia uma `id` de `labecommerce_products`
 - `quantity` : number
 - `total_price` : number
- express:
 - método: `post`
 - path: `/purchases`
 - `id` deve ser gerado pela própria aplicação em código
 - `total_price` deve ser calculado pela própria aplicação em código
 - parâmetros recebidos via `body` :
 - `user_id` , `product_id` e `quantity`

Busca das compras de um usuário

Essa funcionalidade irá permitir a exibição do histórico de compras no perfil do usuário.

- método: `get`
- path: `/users/:user_id/purchases`
- parâmetro recebido via `path params` :
 - `user_id`
- deve trazer uma lista com **todas as compras** de um determinado usuário

