

Faculdade Multivix
Sistemas de Informação
Programação para internet - Projeto Final Parte 1

1 Instruções de entrega

Todos os arquivos gerados nessa atividade devem ser entregues via email, com as orientações a seguir:

- Email : brenokrohling@professor.multivix.edu.br
- Data de entrega: 12/06/2025
- Assunto: PWEB-FINALPT1

2 Atividade

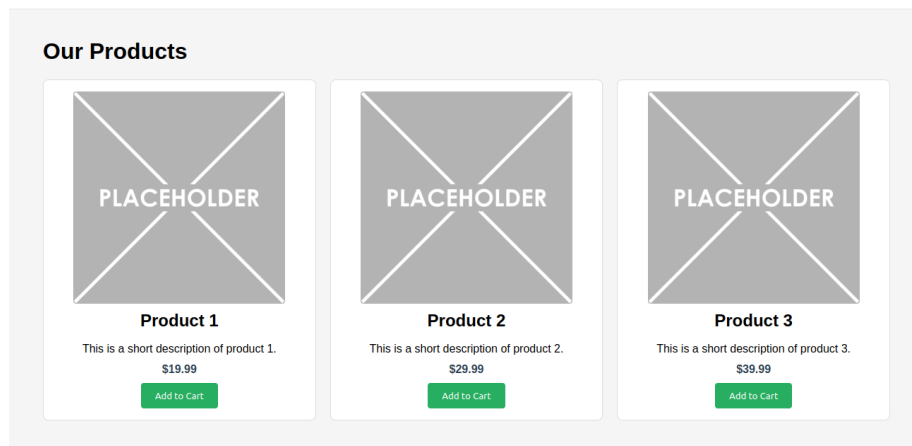
No projeto final de programação para internet iremos simular - de maneira simples - uma loja virtual. Para isso, utilizaremos as tecnologias conhecidas em sala: HTML, CSS, Javascript/jQuery, protocolo HTTP, etc.

Na primeira parte do do projeto temos dois objetivos

1. Criar uma página que lista produtos cadastrados em um servidor
2. Criar uma página que cria novas produtos

2.1 Página de listagem de produtos

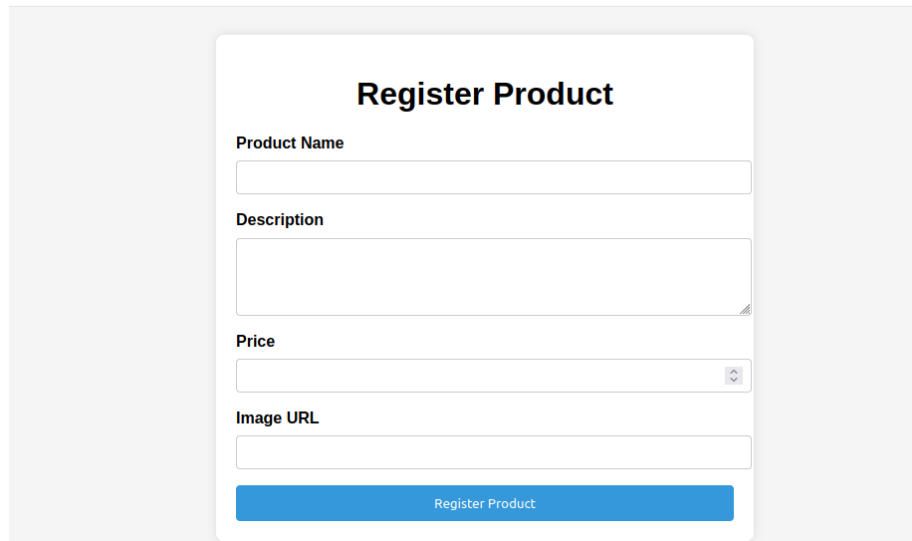
A página de listagem de produtos deve seguir um layout similar ao apresentado abaixo:



Os produtos a serem listados devem ser carregados dinamicamente utilizando funções javascript que farão requests http a um servidor que será disponibilizado para os alunos. A função que faz o request já está implementada, restando apenas finalizar a lógica para que o retorno dos dados seja apresentado na página de forma estruturada.

2.2 Página de cadastro de produto

Deve ser criado uma página que contém um formulário, que ao ser enviado faz uma requisição para criar um novo produto no servidor. A função que realiza tal request está parcialmente implementada, restando passar os dados necessários para realizar o request. O layout da página pode seguir algo como apresentado abaixo:

A screenshot of a web form titled "Register Product". The form is white with a blue button at the bottom. It contains four input fields: "Product Name" (a single-line text box), "Description" (a multi-line text area), "Price" (a single-line text box with a currency symbol icon on the right), and "Image URL" (a single-line text box). The button is blue with the text "Register Product" in white.

3 Execução do Servidor Flask

3.1 Pré-requisitos

Para executar a API desenvolvida com Flask, é necessário ter o seguinte instalado:

- **Python 3.7+**
- **pip** (gerenciador de pacotes do Python)

3.2 Instalação das dependências

Antes de executar o servidor, instale as dependências do projeto com o seguinte comando:

```
pip install flask flask-cors
```

3.3 Estrutura de arquivos esperada

O projeto deve conter pelo menos os seguintes arquivos:

- **app.py** – código principal da API Flask
- **products.json** – arquivo JSON com os dados persistidos dos produtos

3.4 Execução do servidor

Para iniciar o servidor, navegue até o diretório onde está localizado o arquivo **app.py** e execute o seguinte comando:

```
python app.py
```

O servidor será iniciado em:

http://localhost:5000

4 Documentação da API

4.1 GET /api/products

- **Descrição:** Retorna a lista de todos os produtos disponíveis.
- **Método:** GET
- **Endpoint:** /api/products
- **Resposta de Sucesso (200 OK):**

```
{
  "products": [
    {
      "id": 1,
      "name": "Mouse sem fio",
      "description": "Mouse ptico com conex o bluetooth",
      "price": 89.90,
      "image": "/images/mouse.jpg"
    },
    ...
  ]
}
```

4.2 POST /api/products

- **Descrição:** Cria um novo produto.
- **Método:** POST
- **Endpoint:** /api/products
- **Cabeçalho Requisitado:**
 - Content-Type: application/json
- **Corpo da Requisição:**

```
{
  "name": "Cadeira Gamer",
  "description": "Cadeira ergon mica com suporte lombar",
  "price": 749.90,
  "image": "/images/cadeira.jpg"
}
```

- **Resposta de Sucesso (201 Created):**

```
{
  "id": 3,
  "name": "Cadeira Gamer",
  "description": "Cadeira ergon mica com suporte lombar",
  "price": 749.90,
  "image": "/images/cadeira.jpg"
}
```