**1. Exercício 1: A Importância da Documentação Técnica em um Projeto de Software**

A documentação técnica é fundamental para qualquer projeto de software, pois serve como uma referência detalhada e precisa para os desenvolvedores, equipes de suporte e outras partes interessadas. Ela descreve como o sistema foi projetado, como funciona internamente e como ele pode ser mantido ou expandido no futuro. Além disso, a documentação técnica facilita a colaboração entre os membros da equipe e garante que todos os envolvidos compreendam as decisões de design, a arquitetura e as soluções implementadas.

**Exemplo de como a documentação pode ajudar no desenvolvimento futuro**: Imagine que, após algum tempo, um novo desenvolvedor precisa trabalhar em um sistema antigo. Se houver documentação clara e atualizada, ele poderá entender rapidamente o funcionamento do sistema, os principais componentes e os requisitos, evitando retrabalho e acelerando o processo de integração.

**2. Exercício 2: Estrutura de Documentação para uma API Simples de Loja Online**

Aqui está uma sugestão de estrutura de documentação para uma API de gerenciamento de produtos em uma loja online:

**Documentação da API - Loja Online**

**Base URL:** https://api.lojaonline.com/v1

**Autenticação:** Para autenticação, utilize o método Bearer Token no cabeçalho da requisição.

**Endpoints:**

1. **Adicionar Produto** **Método:** POST  
    **URL:** /produtos  
    **Descrição:** Adiciona um novo produto à loja online.  
    **Parâmetros:**
   * nome (string) - Nome do produto (obrigatório)
   * descricao (string) - Descrição do produto
   * preco (decimal) - Preço do produto (obrigatório)
   * quantidade (int) - Quantidade disponível (obrigatório)
2. **Resposta:**
   * 201 Created - Produto criado com sucesso.
3. **Editar Produto** **Método:** PUT  
    **URL:** /produtos/{id}  
    **Descrição:** Edita as informações de um produto existente.  
    **Parâmetros:**
   * nome (string) - Nome do produto
   * descricao (string) - Descrição do produto
   * preco (decimal) - Novo preço do produto
   * quantidade (int) - Nova quantidade disponível
4. **Resposta:**
   * 200 OK - Produto atualizado com sucesso.
5. **Excluir Produto** **Método:** DELETE  
    **URL:** /produtos/{id}  
    **Descrição:** Remove um produto da loja online.  
    **Resposta:**
   * 204 No Content - Produto excluído com sucesso.

**3. Exercício 3: Tutorial para Registrar uma Nova Conta em uma Plataforma de E-commerce**

**Passo a Passo para Registrar uma Conta:**

1. **Acessar o site** Vá até a página inicial da plataforma de e-commerce.
2. **Clicar em "Registrar"** No canto superior direito da página inicial, clique no botão **"Registrar"**.
3. **Preencher os dados pessoais** Preencha os campos obrigatórios com as seguintes informações:  
   * **Nome Completo:** Seu nome completo.
   * **Email:** Um email válido para comunicação.
   * **Senha:** Escolha uma senha segura, com pelo menos 8 caracteres.
   * **Confirmar Senha:** Digite a senha novamente.
4. **Aceitar os Termos e Condições** Marque a caixa de verificação para aceitar os Termos de Serviço e a Política de Privacidade.
5. **Clique em "Cadastrar"** Após preencher os dados, clique no botão **"Cadastrar"**.
6. **Verificação de Email** Você receberá um email de verificação. Abra o email e clique no link para ativar sua conta.
7. **Conta Criada com Sucesso** Após a ativação, você poderá fazer login na plataforma usando seu email e senha.

**4. Exercício 4: Boas Práticas na Escrita de Documentação de Software**

A documentação de software deve ser clara, concisa e bem organizada para garantir que todos os usuários e desenvolvedores possam facilmente entender e utilizar o sistema. Algumas boas práticas incluem:

* **Clareza e Objetividade:** Utilize uma linguagem simples e direta. Evite jargões técnicos, a menos que sejam necessários, e forneça explicações quando isso ocorrer.
* **Estrutura Organizada:** Divida a documentação em seções lógicas, como introdução, instalação, configuração, exemplos de uso, e considerações finais.
* **Exemplos Práticos:** Sempre que possível, forneça exemplos de código ou casos de uso para ilustrar como utilizar os recursos descritos.

**Exemplo:**

"Para criar uma nova instância do serviço de pagamento, utilize o seguinte código:

servico\_pagamento = Pagamento(cliente, valor)

servico\_pagamento.processar()

Isso garantirá que o pagamento seja processado corretamente."

**5. Exercício 5: Diferenças entre Documentação Técnica, de Usuário e de API**

* **Documentação Técnica:** Destinada a desenvolvedores, analistas e equipes de TI. Descreve a arquitetura do sistema, o fluxo de dados, as tecnologias utilizadas e como o software foi implementado. É importante manter essa documentação atualizada, pois ela serve como referência para manutenção, correção de bugs e futuras melhorias.
* **Documentação de Usuário:** Focada em guiar o usuário final na utilização do software. Explica como utilizar os recursos da plataforma e resolve problemas comuns. Manter essa documentação atualizada garante que os usuários tenham uma experiência satisfatória e que possam usar o sistema de maneira eficaz.
* **Documentação de API:** Fornece informações sobre como interagir com os endpoints de uma API. Detalha os métodos HTTP, os parâmetros e as respostas esperadas. Manter essa documentação atualizada é crucial para que os desenvolvedores que utilizam a API saibam como integrá-la corretamente.

**Importância de Manter Cada Tipo de Documentação Atualizada:** Manter cada tipo de documentação atualizada é essencial para garantir que todos os envolvidos no projeto, seja no desenvolvimento, manutenção ou uso do sistema, tenham acesso às informações corretas e atualizadas. Isso previne erros, reduz o tempo de treinamento e acelera o processo de resolução de problemas.