1 - A documentação técnica é fundamental para garantir a continuidade, manutenção e evolução de um projeto de software. Ela descreve o funcionamento interno do sistema, permitindo que desenvolvedores entendam a lógica, arquitetura e fluxos de dados sem depender exclusivamente dos criadores originais.

**Exemplo**:  
Em um sistema legado, se não houver documentação técnica, um novo desenvolvedor levará dias ou semanas para entender a estrutura e realizar correções ou melhorias. Com uma documentação clara, esse tempo é reduzido drasticamente, permitindo manutenção rápida e segura.

--

2 - **1. Introdução**  
Explicação geral sobre a API e seus objetivos.

**2. Autenticação**  
Explicação sobre o processo de autenticação para acessar a API.

**3. Endpoints**  
**3.1. Adicionar Produto**  
Endpoint responsável pela criação de novos produtos.

**3.2. Editar Produto**  
Endpoint responsável pela atualização de produtos existentes.

**3.3. Excluir Produto**  
Endpoint utilizado para a remoção de produtos.

**3.4. Listar Produtos**  
Endpoint que lista todos os produtos cadastrados.

**4. Exemplos de Uso**  
Exemplos práticos de chamadas e respostas da API, utilizando formatos como JSON ou cURL.

**5. Erros Comuns**  
Relação dos erros mais frequentes e orientações de como solucioná-los.

--

3 - **1.** Acesse o site da loja online.  
**2.** Clique no botão "Criar Conta" ou "Registrar-se" no topo da página.  
**3.** Preencha o formulário com as seguintes informações:  
  - Nome completo  
  - E-mail válido  
  - Senha forte  
  - Confirmação da senha  
**4.** Aceite os Termos de Uso e a Política de Privacidade (se solicitado).  
**5.** Clique no botão "Registrar".  
**6.** Acesse seu e-mail e clique no link de ativação enviado pela plataforma.  
**7.** Retorne ao site e realize o login com suas credenciais recém-criadas.

--

4 - **1. Método e Caminho**  
POST /api/products

**2. Descrição**  
Cria um novo produto no sistema de catálogo da loja.

**3. Parâmetros da Requisição**  
name (string): Nome do produto — obrigatório  
description (string): Descrição detalhada — opcional  
price (number): Preço em reais — obrigatório

**4. Exemplo de Requisição**

json

CopiarEditar

{

"name": "Mesa Gamer",

"description": "Mesa ergonômica para jogos e trabalho",

"price": 599.90}

**5. Exemplo de Resposta**

json

CopiarEditar

{

"id": 101,

"name": "Mesa Gamer",

"description": "Mesa ergonômica para jogos e trabalho",

"price": 599.90,

"created\_at": "2025-04-29T14:00:00Z"}

1. **Notas Importantes**  
   O campo "price" deve ser maior que 0.  
   O campo "name" deve ter no máximo 100 caracteres.

--

5 - **Tipo: Documentação Técnica**  
Público-alvo: Desenvolvedores, engenheiros  
Propósito: Explicar a arquitetura e funcionamento do sistema  
Exemplo: Diagramas, fluxos de dados

**Tipo: Documentação de Usuário**  
Público-alvo: Usuários finais  
Propósito: Ensinar como utilizar o software  
Exemplo: Manuais, tutoriais passo a passo

**Tipo: Documentação de API**  
Público-alvo: Desenvolvedores externos  
Propósito: Explicar como consumir e integrar a API  
Exemplo: Endpoints, exemplos de chamadas

**Importância de manter atualizadas:**  
Sistemas mudam constantemente. Documentações desatualizadas causam erros, confusões e até falhas na integração ou uso.

Garantir atualização facilita manutenções, novos desenvolvimentos e aumenta a confiabilidade do sistema para todos os públicos envolvidos.

1 - O MKDocs é uma ferramenta de geração de sites estáticos de documentação técnica, escrita em Markdown. Ele permite criar documentação de forma rápida, organizada e com aparência profissional, sem necessidade de conhecimentos avançados em HTML ou CSS.

**Principais vantagens do MKDocs:**  
– É fácil de instalar e usar  
– Usa Markdown, uma linguagem simples e acessível  
– Possui temas prontos e personalizáveis (como o tema "material")  
– Gera navegação automática entre páginas  
– Permite pré-visualização local com atualização em tempo real  
– Exporta toda a documentação como HTML estático, facilitando o deploy em qualquer servidor

--

5 - Markdown é uma linguagem de marcação simples e leve, que permite criar documentos com formatação básica (como títulos, listas e blocos de código) de forma rápida, legível e sem necessidade de editores complexos.

**Vantagens do Markdown:**

– É fácil de aprender e escrever  
– Pode ser lido diretamente, mesmo sem renderização  
– É compatível com várias plataformas e ferramentas (como GitHub, MKDocs, etc.)

1 - Após criar um projeto com o comando mkdocs new, é gerada uma estrutura básica com os seguintes elementos:

**meu-projeto/**  
Pasta principal do projeto.

**mkdocs.yml**  
Arquivo de configuração central do MKDocs. Nele são definidos o nome do site, o tema, a navegação e os plugins.

**docs/**  
Pasta onde ficam armazenados todos os arquivos de documentação, geralmente escritos em Markdown (.md).

**index.md**  
Página inicial da documentação. Este será o primeiro arquivo exibido quando o site for carregado, funcionando como a "Home".

1. O uso de sistemas de controle de versão como o Git no gerenciamento de documentação técnica é importante porque permite rastrear todas as mudanças feitas, saber quem fez, quando fez e por quê. Além disso, facilita a colaboração entre vários autores, possibilita reverter alterações quando necessário, e mantém um histórico organizado da evolução da documentação.

1 - Instalacao do navegador Google Chrome  
Acesse o site www.google.com/chrome  
Clique no botao Baixar Chrome  
Abra o arquivo baixado chamado ChromeSetup.exe  
Siga as instrucoes na tela  
O Chrome sera iniciado automaticamente

2 - Os usuarios enviam arquivos que ficam salvos no servidor e podem ser acessados pelos administradores depois

3 - Instalacao e configuracao do ambiente Node

Passo 1  
Instalar o Nodejs

Passo 2  
Configurar o ambiente com variaveis de ambiente

Passo 3  
Testar a instalacao com um comando no terminal

Passo 4  
Criar um novo projeto utilizando npm init

4 - A funcao abaixo soma dois numeros e retorna o resultado  
Ela pode ser usada sempre que for necessario calcular a soma de dois valores informados pelo usuario

Exemplo em Python

def somar(a b):  
return a + b

resultado = somar(5 3)  
print(resultado)

5 - Inicio  
-> Usuario digita login e senha  
-> Sistema verifica os dados

Se os dados estiverem corretos  
-> Acesso permitido  
-> Fim

Se os dados estiverem incorretos  
-> Mostrar mensagem de erro  
-> Voltar para a entrada de dados

1 - A documentacao automatizada evita que os desenvolvedores precisem escrever tudo manualmente  
Ela garante que a documentacao esteja sempre atualizada com o codigo real  
Isso reduz erros de informacao e economiza tempo da equipe  
Tambem facilita a manutencao da documentacao porque tudo muda junto com o codigo  
Com isso a equipe trabalha de forma mais rapida organizada e segura

2 - Projeto de documentacao de API com MKDocs e Swagger

-- Criar uma pasta chamada sistema usuarios  
 Dentro da pasta criar um arquivo chamado openapi.yaml com o seguinte conteudo

openapi 3.0.0  
info  
title API de Usuarios  
description API para gerenciar usuarios de um sistema ficticio  
version 1.0.0  
paths  
/usuarios  
get  
summary Lista todos os usuarios  
responses  
200  
description Sucesso

-- Baixar os arquivos do swagger ui e colocar em uma pasta chamada swagger dentro do projeto

-- Criar um arquivo chamado api.md com o seguinte conteudo

API de Usuarios

<iframe src="swagger/index.html" width="100%" height="800"></iframe>

-- Criar o arquivo mkdocs.yml com o seguinte conteudo

site\_name Documentacao do Sistema  
nav

Home index.md

API api.md

Theme

name material

-- Rodar o comando pip install mkdocs  
-- Rodar mkdocs serve para ver o projeto funcionando

3 -