

Trilha 04

Versionamento e Deploy da Documentação

Instruções para a melhor prática de Estudo

1. **Leia atentamente todo o conteúdo:** Antes de iniciar qualquer atividade, faça uma leitura detalhada do material fornecido na trilha, compreendendo os conceitos e os exemplos apresentados.
2. **Não se limite ao material da trilha:** Utilize o material da trilha como base, mas busque outros materiais de apoio, como livros, artigos acadêmicos, vídeos, e blogs especializados. Isso enriquecerá o entendimento sobre o tema.
3. **Explore a literatura:** Consulte livros e publicações reconhecidas na área, buscando expandir seu conhecimento além do que foi apresentado. A literatura acadêmica oferece uma base sólida para a compreensão de temas complexos.
4. **Realize todas as atividades propostas:** Conclua cada uma das atividades práticas e teóricas, garantindo que você esteja aplicando o conhecimento adquirido de maneira ativa.
5. **Evite o uso de Inteligência Artificial para resolução de atividades:** Utilize suas próprias habilidades e conhecimentos para resolver os exercícios. O aprendizado vem do esforço e da prática.
6. **Participe de debates:** Discuta os conteúdos estudados com professores, colegas e profissionais da área. O debate enriquece o entendimento e permite a troca de diferentes pontos de vista.
7. **Pratique regularmente:** Não deixe as atividades para a última hora. Pratique diariamente e revise o conteúdo com frequência para consolidar o aprendizado.
8. **Peça feedback:** Solicite o retorno dos professores sobre suas atividades e participe de discussões sobre os erros e acertos, utilizando o feedback para aprimorar suas habilidades.

Essas instruções são fundamentais para garantir um aprendizado profundo e eficaz ao longo das trilhas.

Versionamento e Deploy da Documentação

Uma das principais vantagens de usar o MKDocs é sua fácil integração com sistemas de controle de versão, como o Git, e sua capacidade de publicar a documentação em plataformas como GitHub Pages. O versionamento é uma prática crucial para garantir que todas as mudanças na documentação sejam rastreadas e que a equipe tenha um histórico claro das modificações. O deploy em plataformas públicas permite que a documentação esteja acessível e atualizada para todos os usuários.

Integração do MKDocs com Sistemas de Controle de Versão (Git)

O Git é uma ferramenta de controle de versão amplamente usada, e a integração com o MKDocs permite gerenciar a documentação de forma organizada. Todo o histórico de mudanças na documentação pode ser rastreado, revertido ou atualizado conforme necessário.

Passos para Integração com Git:

1. **Inicialize o Repositório Git:** Dentro do diretório do projeto MKDocs, inicialize o repositório Git:

```
git init
```

2. **Adicione os Arquivos ao Controle de Versão:** Após inicializar o repositório, adicione os arquivos ao controle de versão:

```
git add .
```

3. **Commit das Alterações:** Faça o commit inicial com a seguinte linha de comando:

```
git commit -m "Adiciona documentação inicial"
```

4. **Criação do Repositório Remoto:** No GitHub, crie um novo repositório para hospedar a documentação. Depois, conecte o repositório local ao remoto:

```
git remote add origin
```

```
https://github.com/seu-usuario/seu-repositorio.git
```

5. **Envio da Documentação para o GitHub:** Finalmente, envie os arquivos para o repositório remoto:

```
git push -u origin main
```

Agora, toda vez que fizer uma alteração na documentação, você pode versionar as mudanças e enviá-las para o repositório GitHub, garantindo que o histórico de mudanças seja preservado.

Publicação de Documentações usando GitHub Pages

O GitHub Pages é uma ferramenta gratuita que permite a hospedagem de sites estáticos diretamente a partir de um repositório GitHub. O MKDocs pode ser facilmente integrado ao GitHub Pages para publicar e compartilhar a documentação com o público.

Passos para Publicação Usando GitHub Pages:

Gerar o Site da Documentação: Antes de publicar, você precisa gerar os arquivos HTML estáticos:

```
mkdocs build
```

1. Isso cria uma pasta chamada `site/` que contém todos os arquivos HTML da documentação.

Deploy Automático Usando o Comando `gh-deploy`: O MKDocs possui um comando embutido que facilita o deploy diretamente no GitHub Pages. Execute o seguinte comando:

```
mkdocs gh-deploy
```

2. Esse comando faz o build da documentação e empurra os arquivos gerados para o branch `gh-pages` do seu repositório GitHub, que é o branch que o GitHub Pages utiliza para publicar o site.

Acessar a Documentação: Após o deploy, você poderá acessar sua documentação no seguinte formato de URL:

```
https://seu-usuario.github.io/seu-repositorio/
```

Observação: Para utilizar o comando `gh-deploy`, certifique-se de que o repositório está público ou que você tem permissões para ativar o GitHub Pages no seu repositório privado.

Atualização Contínua e Manutenção da Documentação

A documentação deve ser mantida atualizada conforme o software evolui. Manter uma rotina de atualização contínua garante que os usuários sempre tenham acesso a informações precisas e relevantes.

Dicas para Manutenção da Documentação:

1. **Commits Frequentes:** Sempre que houver uma alteração significativa na documentação (adição de novas páginas, correção de erros ou atualização de exemplos), faça um commit dessas mudanças.
2. **Revisão Colaborativa:** Utilize pull requests e revisões de código para garantir que as alterações feitas na documentação sejam revisadas e aprovadas por outros membros da equipe.
3. **Deploy Contínuo:** Automatize o processo de deploy para que, sempre que uma alteração seja feita no branch principal da documentação, ela seja automaticamente publicada. Isso pode ser feito com ferramentas como GitHub Actions.
4. **Organização de Versões:** Se o software possuir múltiplas versões, é interessante manter a documentação para cada versão de forma separada. O MKDocs permite organizar isso com a ajuda de plugins como `mkdocs-versioning`.

Exemplos Práticos

Commit e Push da Documentação: Após criar a documentação inicial, siga os comandos abaixo para versioná-la:

```
git init
git add .
git commit -m "Adiciona documentação inicial"
git remote add origin
https://github.com/seu-usuario/seu-repositorio.git
git push -u origin main
```

1. **Deploy Automático com `gh-deploy`:** Após configurar seu repositório GitHub, execute:

```
mkdocs gh-deploy
```

2. Isso publicará sua documentação no GitHub Pages.

Configuração de GitHub Actions para Deploy Contínuo: Crie um arquivo `.github/workflows/deploy.yml` no seu repositório com o seguinte conteúdo:

```
name: Deploy Documentation
on:
  push:
    branches:
      - main
jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v2
      - name: Set up Python
        uses: actions/setup-python@v2
        with:
          python-version: '3.x'
      - name: Install dependencies
        run: |
          pip install mkdocs mkdocs-material
      - name: Deploy to GitHub Pages
        run: |
          mkdocs gh-deploy --force
```

3. Isso automatiza o deploy da documentação sempre que houver uma mudança no branch `main`.

Lista de Exercícios de Fixação

1. **Exercício 1:** Explique a importância de utilizar sistemas de controle de versão, como Git, no gerenciamento de documentação técnica.
2. **Exercício 2:** Crie um repositório Git para seu projeto MKDocs e faça o commit e push da documentação inicial.
3. **Exercício 3:** Realize o deploy da sua documentação no GitHub Pages utilizando o comando `mkdocs gh-deploy`. Após a publicação, acesse o link gerado e verifique se a documentação está acessível.
4. **Exercício 4:** Configure um sistema de deploy contínuo para a documentação utilizando GitHub Actions. Teste se a documentação é atualizada automaticamente ao fazer mudanças no repositório.
5. **Exercício 5:** Pesquise sobre como organizar múltiplas versões da documentação com MKDocs e implemente uma solução que permita ao usuário alternar entre versões.