

Trilha 10

*Otimizando e Debugando
Contêineres*

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.

Passo 10: Otimizando e Debugando Contêineres

Conceito:

A otimização e depuração de contêineres garantem eficiência e resolução de problemas. Docker oferece ferramentas como logs e shells interativos.

Exemplo:

1. Veja os logs de um contêiner:

```
docker logs <container_id>
```

2. Acesse o shell interativo do contêiner:

```
docker exec -it <container_id> sh
```

Exercícios de fixação:

- Depure um contêiner com erro.
 - Otimize um Dockerfile para reduzir o tamanho da imagem.
-

Desafios

Desafio 1: Criar uma imagem Docker contendo PHP, MySQL, HTML e CSS rodando em Nginx e Composer

Passo a Passo:

1. Crie um **Dockerfile** que inclua PHP e MySQL.
 2. Configure o **nginx.conf** para rodar o servidor web.
 3. Adicione Composer no **Dockerfile**.
 4. Crie um **docker-compose.yml** para orquestrar os serviços.
-

Desafio 2: Criar uma imagem Docker contendo VueJS, NodeJS, MySQL, rodando em Nginx e Composer

Passo a Passo:

1. Crie um **Dockerfile** que inclua NodeJS e VueJS.
 2. Configure o **nginx.conf** para servir a aplicação.
 3. Adicione MySQL e Composer para gerenciar dependências.
 4. Crie um **docker-compose.yml** para orquestrar os serviços.
-

Referências:

- Documentação oficial do Docker: [Docker CLI - Contêiner](#)
- Imagens Docker: [Docker Hub](#)