



Hoje veremos com mais detalhes o comando **WORKDIR** do Docker.

O comando **WORKDIR** é usado para definir o diretório de trabalho de um contêiner do Docker a qualquer momento.

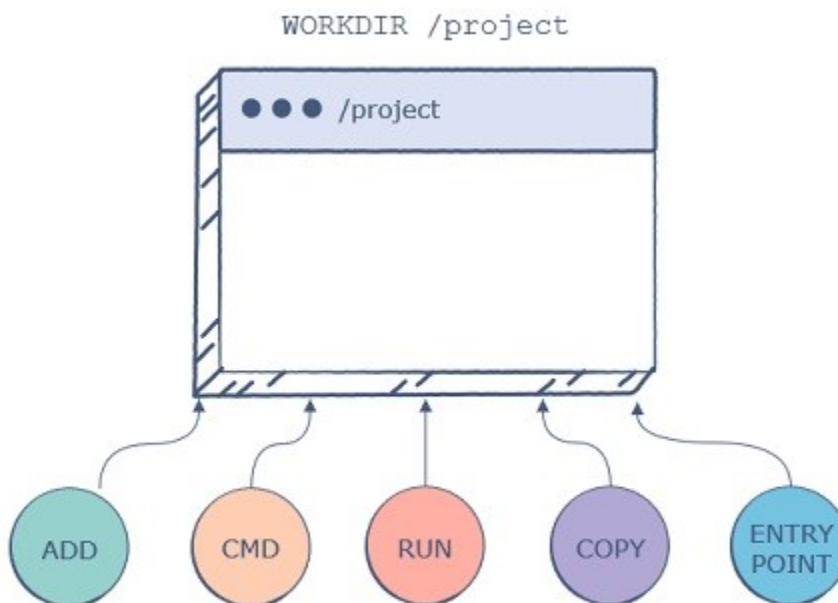
O comando é especificado no arquivo **Dockerfile**.



Curso Docker Essencial para .NET

Obs: Um arquivo **Dockerfile** é o arquivo de texto que cria **imagens** do Docker. Este arquivo de texto contém instruções específicas sobre o contêiner.

Qualquer comando **RUN**, **CMD**, **ADD**, **COPY** ou **ENTRYPOINT** será executado no diretório de trabalho especificado pelo comando **WORKDIR**.



Se o comando **WORKDIR** não estiver gravado no **Dockerfile**, ele será criado automaticamente pelo compilador Docker.

Portanto, pode-se dizer que o comando executa **mkdir** e **cd** implicitamente:

FROM ubuntu:16.04

WORKDIR /projeto

RUN npm install

Neste código se o diretório do **projeto** não existir, ele será criado e o comando **RUN** será executado dentro do diretório **projeto**.

Nota: O comando **WORKDIR /usr/src/app** seria então equivalente a executar **RUN mkdir -p /usr/src/app**

Reutilizando WORDDIR

O comando **WORKDIR** pode ser reutilizado para definir um novo diretório de trabalho em qualquer estágio do **Dockerfile**.

O caminho do novo diretório de trabalho deve ser fornecido em relação ao diretório de trabalho atual.

FROM ubuntu:16.04

WORKDIR /projeto

RUN npm install

WORKDIR ../projeto2

RUN touch arquivo.cpp

Embora os diretórios possam ser criados e alterados manualmente, é altamente recomendável que você use o **WORKDIR** para especificar o diretório atual no qual você gostaria de trabalhar, pois isso facilita a solução de problemas.

Desta forma você pode pensar no **WORKDIR** como um CD dentro do contêiner (*ele afeta os comandos que vêm depois no Dockerfile, como o comando RUN*).

Se você removeu o **WORKDIR** no seu exemplo acima, a instalação do **RUN npm** não funcionaria porque você não estaria no diretório **/projeto** dentro do seu contêiner.

A seguir temos um exemplo onde aplicação do comando **WORKDIR** é feita no local errado :

FROM microsoft/aspnetcore:2

COPY --from=build-env /publish /publish

WORKDIR /publish

ENTRYPOINT ["dotnet", "/publish/api.dll"]

Vamos corrigir a aplicação do comando WORKDIR :

FROM microsoft/aspnetcore:2

WORKDIR /publish

COPY --from=build-env /publish

ENTRYPOINT ["dotnet", "/publish/api.dll"]

No código também removemos a indicação para a pasta `/publish` do comando `COPY`.

E estamos conversados... 😊

"E, chegando-se Jesus, falou-lhes, dizendo: É-me dado todo o poder no céu e na terra."
[Mateus 28:18](#)



Visite a loja virtual e encontre

Cursos e recursos de aprendizagem para
a plataforma .NET

Referências:

- [Seção Visual Basic do site Macoratti .net](#)
- [C# - StreamReader, StringReader e TextReader .](#)
- [VB 2005 Streams - Macoratti.net](#)
- [C# - Imprimindo um arquivo texto -](#)
- [Serialização de objetos - Macoratti.net](#)

- [C# - Usando OpenFileDialog - Macoratti](#)
- [Compactando e Descompactando arquivos](#)
- [Docker - Criando um Contêiner para .NET Core](#)
- [Minicurso Docker - Apresentando o Docker](#)
- [Docker - Trabalhando com contêineres](#)
- [Docker - Uma introdução básica](#)

[José Carlos Macoratti](#)