

# Docker

## 1 Converter aplicação para imagem

### 1.1 Dockerfile

```
FROM <plataforma>:<sistema_operacional>
COPY <origem> / < destino>
WORKDIR <diretorio_trabalho>
CMD <comando_run>
```

#### 1.1.1 Construindo imagem

```
#construindo diretorio local
docker build -t <nome_imagem>
#vizualizando as imagens
docker images
```

## 2 Application

### 2.1 Adicionando sistemas operacionais

```
FROM node:12- alpine #adiciona alinguagem e o sistema operacional
WORKDIR /App #Especificacao direroiro de trabalho
COPY .. # Copia todo diretorio
```

### 2.2 Comando RUN

```
FROM node:12-alpine #image do docker hub

WORKDIR /app # diretorio do image

COPY . . # copia dos arquivos

ADD https://microsoft.com/teste.json .% copia dos arquivos online
compactados ou descompactados
RUN apk add --no-cache python2 g++ make % adiconando python no alpine
linux
CMD [ ]
```

**RUN apk add --no-cache python2 g++ make**

- apk: é o utilitário de gerenciamento de pacotes do Alpine Linux.
- add: é o subcomando utilizado para adicionar pacotes.
- --no-cache: é uma opção que instrui o apk a não salvar o índice localmente após a instalação, o que economiza espaço em disco, mas também significa que não será possível atualizar ou reinstalar pacotes sem uma conexão com a internet.
- python2: é o pacote do Python 2 que será instalado.
- g++: é o compilador C++.
- make: é uma ferramenta de automação de compilação utilizada principalmente para compilar e construir projetos de software