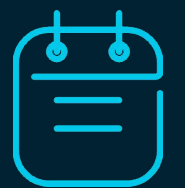




Start <tech>

MiniCurso: Configurando WSL,
Docker e VS Code para Projeto
File Storage App



Jul/202
5

Objetivo do MiniCurso

Configurar ambiente de desenvolvimento moderno no Windows abordando algumas tecnologias como:

==Backend==

Node.js + Express

PostgreSQL

MinIO (compatível com S3)

Multer

==Frontend:==

HTML, CSS, JavaScript

==Infraestrutura==

Docker / Docker Compose

WSL/PgAdmin



História do Docker

Como tudo começou:

Criado pela empresa dotCloud (posteriormente renomeada para Docker Inc.).

Lançado como projeto open source em 2013. Inspirado em tecnologias como LXC (Linux Containers).

Popularizou o conceito de "containers" como alternativa leve às máquinas virtuais.

Hoje é padrão de mercado para empacotamento e distribuição de aplicações.



O que é Docker?

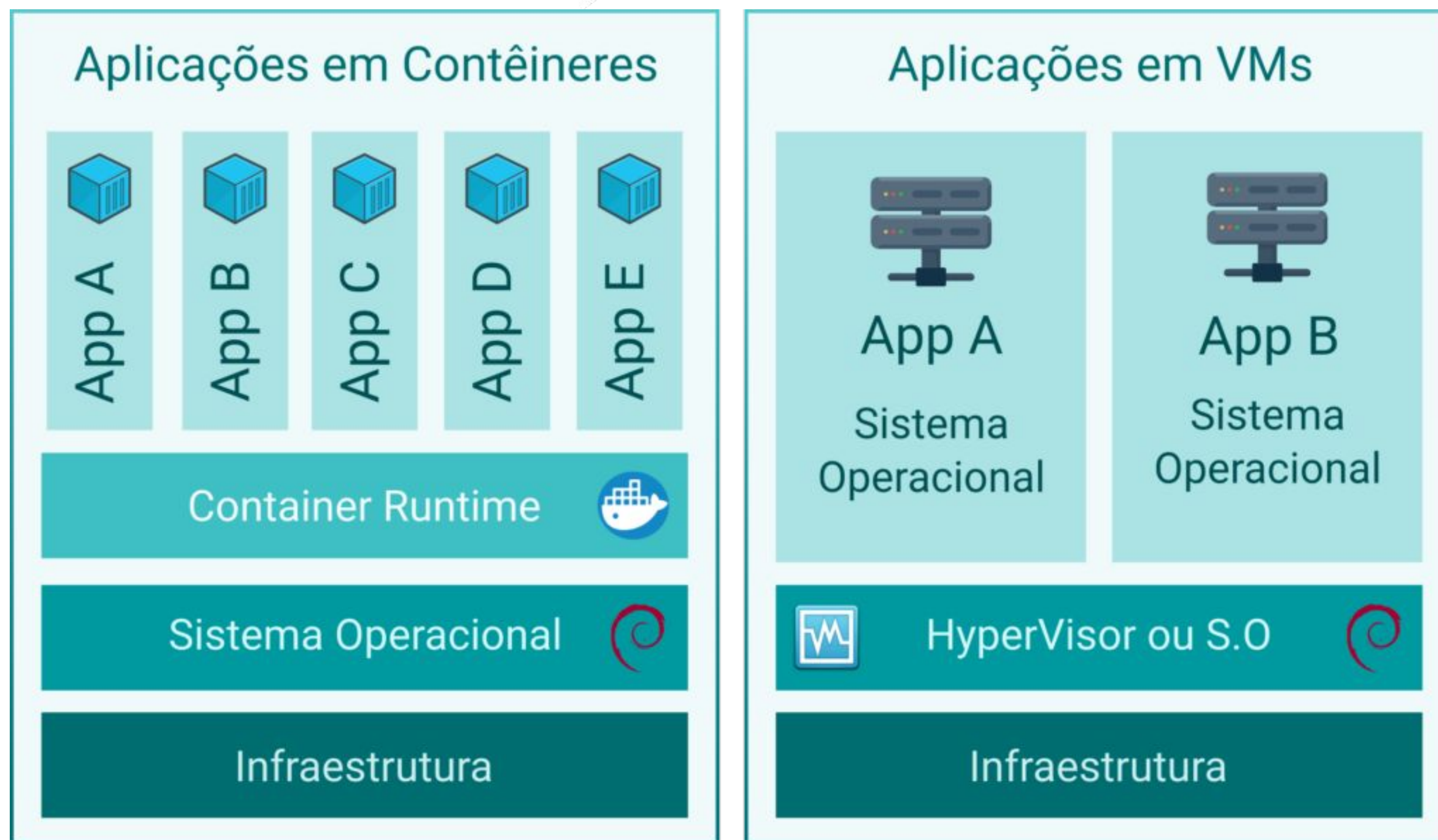
Como tudo começou:

Plataforma para desenvolver, empacotar e executar aplicações em containers.

Containers são leves, portáteis e consistentes.

Evita problemas do tipo "na minha máquina funciona".

Permite criar ambientes isolados para cada aplicação.



O que são Imagens Docker?

5



Imagem é um snapshot imutável de uma aplicação + dependências.

Criada com instruções de um Dockerfile.

Pode ser compartilhada e reutilizada.

Base para a criação de containers.



Preparar Ambiente Windows



Este guia apresenta os passos para configurar um ambiente de desenvolvimento no Windows usando WSL2 e Docker Engine nativo.

1. Instalar WSL2

1.1. Habilitar recursos necessários do Windows

Execute no PowerShell em modo administrador:

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

1.2. Reiniciar o Windows

Após habilitar os recursos, reinicie o computador.

1.3. Instalar o Kernel do WSL2

Baixe e instale o pacote de atualização do kernel do WSL2: [WSL2 Kernel Update](#)

1.4. Definir WSL2 como versão padrão

Execute no PowerShell:

```
wsl --set-default-version 2
```

2. Instalar Ubuntu

Há duas opções para instalar o Ubuntu:

Opção 1 - Via Microsoft Store (Recomendado)

1. Acesse a [Microsoft Store](#)

2. Instale o Ubuntu (versão sem número)

3. Inicie o Ubuntu e configure seu usuário e senha

Preparar Ambiente Windows



Opção 2 - Via linha de comando

Execute no PowerShell:

```
wsl --install -d Ubuntu
```

3. Instalar Docker Engine no Ubuntu

1. Atualize o sistema e instale dependências:

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
```

```
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

2. Adicione o repositório do Docker:

```
echo \
```

```
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
```

```
https://download.docker.com/linux/ubuntu \
```

```
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Preparar Ambiente Windows



3. Instale o Docker Engine e ferramentas:

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

4. Adicione seu usuário ao grupo docker:

```
sudo usermod -aG docker $USER
```

5. Reinicie o WSL:

```
# Execute no PowerShell do Windows
```

```
wsl --shutdown
```

6. Configure o Docker para iniciar automaticamente:

```
# Edite o arquivo wsl.conf
```

```
sudo vim /etc/wsl.conf
```

```
# Adicione o conteúdo (pressione i para inserir):
```

```
[boot]
```

```
command = service docker start
```

```
# Salve e saia (pressione Esc, digite :wq e pressione Enter)
```

4. Instalar Windows Terminal

1. Acesse [Windows Terminal](#) na Microsoft Store

2. Clique em Instalar

Preparar Ambiente Windows



5. Configurações Recomendadas

5.1. Configurar limites do WSL2

Crie/edite o arquivo `.wslconfig` na pasta do seu usuário Windows (`C:\Users\<seu_usuario>\.wslconfig`):

```
[wsl2]
```

```
memory=8GB
```

```
processors=4
```

```
networkingMode=mirrored
```

```
[experimental]
```

```
autoMemoryReclaim=gradual
```

```
sparseVhd=true
```

5.2. Otimizar Docker Build

Adicione ao final do arquivo `~/.profile` no Ubuntu:

```
export DOCKER_BUILDKIT=1
```

Verificação da Instalação

1. Reinicie o WSL:

```
wsl --shutdown
```

2. Abra o Windows Terminal e inicie o Ubuntu

VS Code com WSL



1. Instalar VS Code ([\[https://code.visualstudio.com\]](https://code.visualstudio.com)(<https://code.visualstudio.com>))

2. Instalar extensão ****Remote - WSL****

3. Abrir projeto com:

```
wsl
```

```
cd ~/projeto
```

```
code .
```

1. Clonar repositório:

```
git clone https://github.com/ghost-linux/starttech_totvs.git
```

```
cd starttech_totvs
```

2. Build e subir containers:

```
docker-compose build
```

```
docker-compose up -d
```

Parar os containers:

```
docker-compose down
```

Parar e remover volumes e containers órfãos:

```
docker-compose down --volumes --remove-orphans
```




Pré-requisitos



Node.js e npm instalados

PostgreSQL em execução

MinIO em execução e acessível na rede

⚠ Atenção: Configuração do arquivo hosts

Para que os links de compartilhamento funcionem corretamente, é necessário mapear o IP do servidor MinIO para o nome minio no arquivo hosts de cada máquina cliente.

Exemplo de entrada no arquivo hosts:

```
192.168.1.12  minio
```

Substitua 192.168.1.12 pelo IP real do servidor MinIO na sua rede.

Windows: C:\Windows\System32\drivers\etc\hosts

Linux/Mac: /etc/hosts

Como editar o arquivo hosts no Linux:

```
sudo vim /etc/hosts
```

Adicione a linha:

```
192.168.1.12  minio
```

No Vim, salve com Esc, digite :wq e pressione Enter.



Acesso aos Serviços



Frontend: <http://localhost:3001>



Backend (API): <http://localhost:3000>



PgAdmin: <http://localhost:5050>

(login: admin@admin.com / senha: admin)



MinIO Console: <http://localhost:9001>

(login: minioadmin / senha: minioadmin)



Como registrar o servidor no PgAdmin

13



Para acessar o banco de dados via PgAdmin, siga os passos abaixo:

Acesse <http://localhost:5050>

Faça login com:

Usuário: admin@admin.com

Senha: admin

Clique com o botão direito em "Servers" > "Register" > "Server..."

Na aba General:

Name: starttech_totvs

Na aba Connection:

Host name/address: starttech_totvs_db_1

Port: 5432

Username: user

Password: password

Marque a opção "Save Password"

Clique em "Save" para concluir.



Consulta SQL de exemplo



Após registrar o servidor, selecione o banco file_storage e execute a seguinte query para visualizar os arquivos cadastrados:

```
SELECT * FROM files;
```


Obrigado(a)



Luiz Herminio

SRE-Linux

luiz.herminio@totvs.com.br

www.linkedin.com/in/luiz-herminio-710810a0

https://github.com/ghost-linux/starttech_totvs



#SOMOSTOTVS

