



DOCKER PARA DESENVOLVEDORES E FERRAMENTAS PORTÁTEIS

Guia Oficial iT Guys

Código: ITGCLI 0003/26 | **Classificação:** PÚBLICO

Responsável: João Pedro Toledo Gonçalves | **Data:** 25/01/2026

1. HISTÓRICO DE REVISÃO

NOTA

REGRA DE OURO:

NOTA

1. **Autor:** João Pedro Toledo Gonçalves.

NOTA

2. **Descrição:** Criação do documento.

Data	Versão	Descrição	Autor
25/01/2026	1.0	Criação Inicial	João Pedro Toledo Gonçalves

2. OBJETIVO

Ensinar o desenvolvedor a usar o Docker para rodar ferramentas, bancos de dados e linguagens sem "sujar" o sistema operacional principal, resolvendo o problema de "Na minha máquina funciona".

3. PRÉ-REQUISITOS

[] Docker Desktop (Windows/Mac) ou Engine (Linux) instalado.

4. CONCEITOS (POR QUE ISSO EXISTE?)

- * **Efêmero (`rm`):** O container nasce, executa e se autodestrói. Zero lixo no disco.
- * **Portátil:** Se roda no seu Docker, roda no servidor de produção.

5. PASSO A PASSO (EXECUÇÃO)

Cenário A: Rodando Ferramentas sem Instalar Nada

NOTA

Imagine que você precisa testar um script Python ou Node.js, mas não quer instalar versões específicas na sua máquina.

1. Rode um script Python (One-off):

```
# Monta a pasta atual (.) em /app e roda o script
docker run --rm -v $(pwd):/app -w /app python:3.9 python meu_script.py
```

- * ``rm``: Remove o container ao terminar.
- * ``-v``: Disponibiliza seus arquivos locais dentro do container.

2. Acesse um terminal Node.js limpo:

```
docker run --rm -it node:18 /bin/bash
```

Cenário B: Banco de Dados de Desenvolvimento

NOTA

Suba um banco Postgres em segundos, use, e destrua.

1. Comando rápido (Oneliner):

```
docker run --rm --name meupostgres -e POSTGRES_PASSWORD=secret -p 5432:5432 postgres:15
```

- * Acesse via DBeaver/PgAdmin em ``localhost:5432``.

Cenário C: Ambiente Completo com Compose

NOTA

A melhor forma de padronizar o time. Crie um ``docker-compose.yml`` na raiz do projeto.

```
version: '3.8'
services:
  app:
    build: .
    ports: ["3000:3000"]
    volumes:
      - ../code # Hot-reload: altera no host, reflete no container
  db:
    image: postgres:15
    environment:
      POSTGRES_PASSWORD: root
    ports: ["5432:5432"]
```

1. Para iniciar tudo:

```
docker compose up
```

6. SOLUÇÃO DE PROBLEMAS (TROUBLESHOOTING)

Problema 1: "Works on my machine" ainda acontece

- * **Causa:** Você está usando volumes (`-v`) que trazem arquivos de config do seu PC para dentro do container.
- * **Solução:** Use `.dockerignore` para não copiar arquivos `node\_modules` ou `.env` locais durante o build.

Problema 2: Porta ocupada

- * **Causa:** Você já tem um Postgres instalado no Windows Services.
- * **Solução:** Pare o serviço local ou mude a porta do container: `-p 5433:5432`.

7. VALIDAÇÃO FINAL

- [] Consegue rodar `docker run rm hello-world` e ele some depois (`docker ps -a`)?
- [] Seu time consegue rodar `docker compose up` e ter o ambiente rodando sem configurar nada?