MANUAL DE ENGENHARIA • 2026

DOCKER NA PRÁTICA PARA PRODUÇÃO

Do ambiente local a serviços seguros, reproduzíveis e observáveis

AUTORIA & DIREÇÃO

O Precursor

BIBLIOTECA O PRECURSOR

Docker na Prática para Produção
Do ambiente local a serviços seguros, reproduzíveis e observáveis

© 2026 O Precursor · BIBLIOTECA O PRECURSOR. Todos os direitos reservados.

Nenhuma parte desta publicação pode ser reproduzida, armazenada ou transmitida de qualquer forma ou por qualquer meio sem a autorização prévia e expressa do titular dos direitos autorais.

Texto, projeto didático e direção editorial: O Precursor. Edição de estudo — 2026.

Mapa do laboratório

Do Dockerfile ao deploy com segurança e recuperação

01. Do notebook ao ambiente reproduzível 8

Um programa que roda não é um sistema pronto 8
Contêineres não são máquinas virtuais completas 9
O laboratório que acompanha o livro 10
Preparação: confira o host antes de fazer qualquer alteração 12
Primeira inicialização controlada 12
Prova do worker e da persistência 14
O que o operador precisa ser capaz de explicar 15
Exercício e critério de aceite 15

02. Dockerfile profissional e imagens enxutas 17

A imagem é um artefato, não uma cópia do notebook 17
Um Dockerfile em dois estágios 18
A ordem das instruções afeta o cache 19
O contexto de build também é uma fronteira 20
Menor privilégio e imutabilidade parcial 21
Tags, digests e cadeia de fornecimento 21
Laboratório: build, inspeção e comparação 22
Caso de engenharia: funciona como root, quebra em produção 23
Exercício e critério de aceite 23

03. Docker Compose para vários serviços 25

A aplicação real não é um único processo 25
A rede é o primeiro desenho de arquitetura 26
Serviço iniciado ainda pode estar indisponível 27
Persistência: volume não é backup 27
Configuração não é segredo 28
Limites de recursos e logs 29
Laboratório de inspeção 30
Revisão arquitetural 30

Figura 1 — Topologia didática: borda, aplicação, fila, worker e banco.

Em vez de oferecer comandos soltos, vamos acompanhar a **Oficina Atlas**, serviço fictício que contém cinco componentes: proxy Caddy, aplicação web Flask, PostgreSQL, Redis Streams e um worker. O objetivo não é ensinar Flask como linguagem principal; é produzir um ambiente concreto para experimentar contêineres, persistência, fila e falhas.

```
Navegador local → Caddy → web → PostgreSQL
                            |
                            +→ Redis Streams → worker
                                                 |
                                                 +→ PostgreSQL
```

O arquivo completo do laboratório está sob a pasta lab/oficina-atlas do projeto do Studio. O proxy publica **somente** a porta 18080 na interface 127.0.0.1. Banco, Redis e worker não possuem portas publicadas. O laboratório não tem autenticação pública, gestão de usuários ou garantias financeiras: portanto, não deve ser colocado na internet sem várias camadas adicionais de engenharia.

O serviço web dispõe de dois endpoints: /health/live, que indica que o processo HTTP está vivo; e /health/ready, que confirma conectividade básica com banco e Redis. Uma API também recebe trabalhos do tipo ping e publica eventos numa fila. O worker os processa de maneira assíncrona e registra um resultado idempotente no banco.

1. Porta, interface, DNS e rede são coisas diferentes

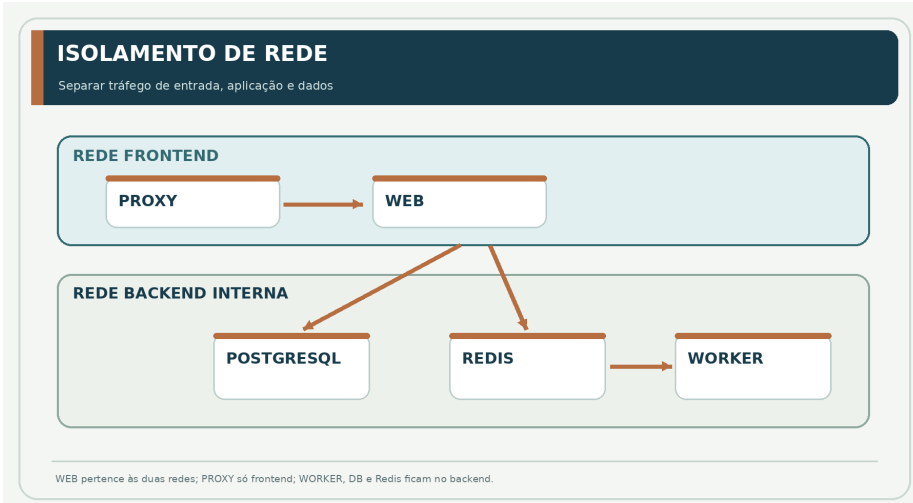


Figura 2 — Redes frontend e backend: limites de comunicação.

Porta de escuta é onde o processo recebe conexões. **Mapeamento de porta** é o vínculo entre host e contêiner. **DNS** transforma nomes em endereços. **Rede Docker** define quais contêineres podem se comunicar. Confundir essas camadas produz erros difíceis de diagnosticar.

Uma configuração como 127.0.0.1:18080:80 publica a porta 80 do proxy apenas na interface de loopback do host, sob a porta 18080. Já 0.0.0.0:80:80 permite escuta em todas as interfaces relevantes. Isso não significa necessariamente que toda conexão externa chegará ao processo: firewall do host, regras da nuvem e topologia de rede também importam.

O Docker Compose cria DNS interno por nome de serviço. O Caddyfile pode usar o hostname web no destino do proxy:

1. Os quatro portões de uma release

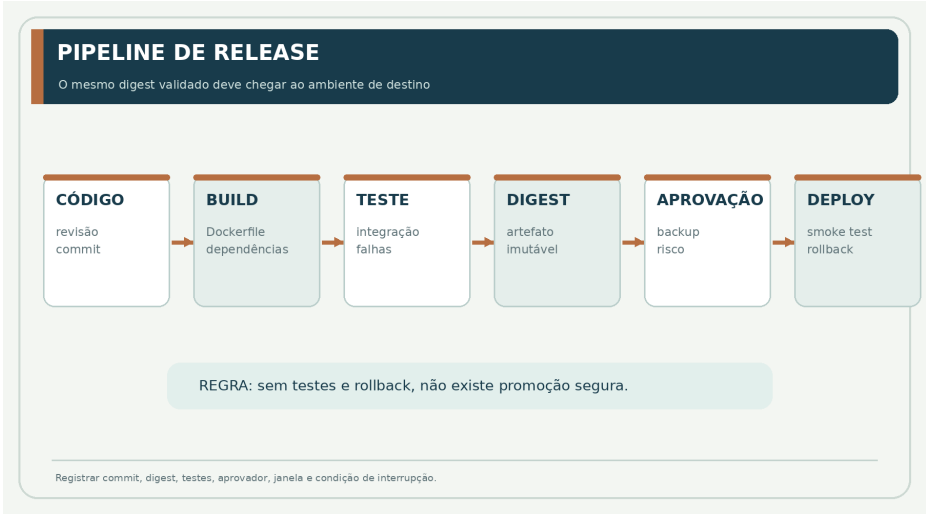


Figura 4 — Build, testes, digest e promoção revisada.

Portão 1 — Fonte: a alteração passou por revisão, o commit está definido, arquivos sensíveis não entraram no repositório e dependências foram analisadas.

Portão 2 — Build: o Dockerfile gera uma imagem com sucesso, a configuração do ambiente é válida e o processo não exige privilégios desnecessários.

Portão 3 — Testes: a imagem é usada em testes de API, integração com banco e fila, permissões, healthchecks e casos de falha. O resultado é armazenado com a referência do build.

Portão 4 — Promoção: um aprovador confere o digest exato, estado da infraestrutura, backup, migrations e procedimento de rollback. A produção recebe o artefato aprovado, não um pacote improvisado.

Apêndice E — Leitura e referências

Documentação oficial, que deve ser consultada novamente na implantação para confirmar sintaxe e versões:

- Docker Engine, imagens, containers, networking, storage e Compose: https://docs.docker.com/
- Dockerfile e boas práticas de build: https://docs.docker.com/build/building/best-practices/
- Redis Streams e consumer groups: https://redis.io/docs/latest/develop/data-types/streams/
- PostgreSQL Backup and Restore: https://www.postgresql.org/docs/current/backup.html
- Caddy reverse proxy, HTTPS e arquitetura: https://caddyserver.com/docs/
- OpenTelemetry e observabilidade: https://opentelemetry.io/docs/

Encerramento: a pergunta final

Um sistema confiável não é aquele que nunca falha; é aquele cujo comportamento pode ser explicado, testado e recuperado sob condições previsíveis. Use o laboratório para formar hábitos de engenharia: hipótese antes de mudança, evidência depois da mudança, proteção de dados antes de operações destrutivas e documentação que outra pessoa consiga executar sem depender da sua memória.



SOBRE O AUTOR

O Precursor

O Precursor é autodidata em tecnologia e desenvolve projetos de software, infraestrutura Linux, automação e inteligência artificial aplicada.

Seu método é construir, testar, documentar e melhorar sistemas com evidências. Neste livro, aplica essa experiência à operação de contêineres, segurança, backups e confiabilidade.

Conecte-se com O Precursor pelos canais oficiais abaixo.

xmarketi.com

Instagram: @0precursor