

# Shiny em Produção: dockerize seu app com 1 linha

Escrito por [Caio](#) em [Tutoriais](#)

13 de setembro de 2019

[docker](#) [shiny](#)

Recentemente, o Julio discutiu aqui no blog [como usar](#) o pacote [golem](#) para facilitar o desenvolvimento de shiny apps em forma de pacotes e [como transformar](#) esses pacotes em "executáveis" de apenas uma linha. Nesse post eu vou continuar essa série e falar sobre como pegar esse app de uma linha e embrulhá-lo em um contêiner docker para que o seu deploy seja instantâneo.

## Inst

Essencialmente toda a magia da transformação de um app em um docker ocorre na pasta `/inst`. Esse diretório, pouco utilizado no dia-a-dia, é útil quando precisamos adicionar dados, templates ou qualquer conteúdo que pode ser necessário em algum momento para o pacote e, portanto, devem ser facilmente acessíveis.

Quando você transformar seu shiny app em pacote utilizando o tutorial do Julio, você terá uma estrutura de diretórios padrão na raiz do seu app: pastas `/R`, `/man` e assim por diante. Crie a pasta `/inst` e dentro dela `/app`; aqui é onde seu app irá morar.

Em princípio, você só precisa de um arquivo `app.R` dentro de `/inst/app` contendo aquela linha mágica que executa seu app todo:

```
shiny::shinyApp(meuParamote:::app_ui(), meuPacote:::app_server)
```

Se você estiver desenvolvendo um app mais complexo, talvez você precise criar também uma pasta `/www` ou um [arquivo\\_auth0.yml](#), mas o resumo da ópera é que você deve ser capaz de rodar `shiny::runApp()` nesta pasta e ver o seu app funcionando perfeitamente.

Durante o desenvolvimento, não se esqueça de **sempre reinstalar** o seu pacote para que `meuParamote:::app_ui()` e `meuParamote:::app_server` mantenham-se atualizados!

O último arquivo que você pode querer adicionar a `/inst/app` é a configuração do seu servidor shiny. No meu caso, eu desenvolvo apps dockerizados para subí-los em máquinas virtuais na nuvem que podem ser acessadas por qualquer um, então preciso deixar clara qual porta deve ser utilizada pelo meu shiny. Como minhas máquinas suportam o protocolo HTTP, meu `shiny-server.conf` é o seguinte:

```
run_as shiny;

# Log all Shiny output to files in this directory
log_dir /var/log/shiny-server;

# Define a server that listens on port 80
server {
    listen 80;

    # Define a location at the base URL
    location / {
        # Host the directory of Shiny Apps stored in this directory
        site_dir /srv/shiny-server;

        # When a user visits the base URL rather than a particular application,
        # an index of the applications available in this directory will be shown.
        directory_index off;
    }
}
```

Fazer com que o app seja ouvido na porta 80 permite que outras pessoas o acessem sem precisar especificar uma porta no URL! Sendo assim, você pode disponibilizar seu shiny app em `meuApp.dominio.com.br`.

## Dockerfile

Agora que a sua pasta `/inst` está devidamente configurada, você precisa criar um `Dockerfile` na raiz do seu app. Isso irá incomodar o `devtools::check()`, mas basta adicionar o nome desse arquivo ao seu `.Rbuildignore` para que ele seja ignorado durante a verificação.

Se todas as dependências do seu aplicativo estiverem devidamente organizadas no `DESCRIPTION` e todas as suas bases internas estiverem propriamente exportadas como documentado pelo Julio, então o seu `Dockerfile` deve ser parecido com o seguinte:

```
FROM rocker/shiny-verse

# Instalar bibliotecas para o tidyverse
RUN apt-get update -qq && apt-get -y --no-install-recommends install \
  build-essential \
  libcurl4-gnutls-dev \
  libxml2-dev \
  libssl-dev \
  r-cran-curl \
  r-cran-openssl \
  curl \
  gnupg1 \
  r-cran-xml2

# Instalar seu próprio app (e suas dependências)
COPY ./ /tmp/app/
RUN R -e "remotes::install_local('/tmp/app/')"

# Copiar arquivos para o lugar certo
EXPOSE 80/tcp
RUN rm /srv/shiny-server/index.html
COPY ./inst/app /srv/shiny-server/
COPY ./inst/app/shiny-server.conf /etc/shiny-server/shiny-server.conf

# Run
CMD ["/usr/bin/shiny-server.sh"]
```

## Deploy

Agora você já pode construir sua imagem docker e executá-la ou subí-la para alguma máquina virtual. Para testar localmente o app, execute os comandos a seguir no terminal e acesse **localhost:8080** no navegador:

```
docker build -t meuApp pasta/para/meuApp
docker run -p 8080:80 meuApp
```

TAMBEM NO CURSO-R.COM

Autenticação em Shiny Apps co

Autenticação em Shiny Apps com o pacote ...

há 2 anos • 2 COMENTÁRIOS

Alguns meses atrás o Daniel fez um post demonstrando como colocar ...

Abaixo ao rvest!

Abaixo ao rvest!

há um ano • 1 COMENTÁRIO

Se você estiver interessado em web scraping, considere participar do nosso ...

Como fazer o R avisar pelo tele

Como fazer o R avisar pelo telegram que ...

há 3 anos • 1 COMENTÁRIO

O ponto máximo da história é o R mandando mensagem pro Telegram. Só para ...

Precisamos falar sobre a partici

Precisamos falar sobre a participação de ...

há 2 anos • 3 COMENTÁRIOS

Introdução Esse ano, vamos participar da Escola de Modelos de Regressão ...

Verificando afi

Verificando com dados

há 2 anos • 2

Uma forma p participar soc utilizando da

### POSTS MAIS RECENTES



## What do you think? Manipulação de dados usando o pacote data.table

Por Fernando Corrêa em [Tutoriais](#) 0 Responses

Upvote

Funny

Love

Surprised

Angry

Disqus' Privacy Policy

[Iniciar sessão](#)

Partilhar

Mostrar primeiro os mais votados



de o seu comentário...

[INICIE SESSÃO COM O](#)

[OU REGISTE-SE NO DISQUS](#)

# BASE

# DOS

# DADOS

DISQUS



[Acessand](#)  
[bases](#)  
[públicas e](#)  
[R: o paco](#)  
[basedosd](#)

Por Fernando ( em [Tutoriais](#)



## Buscando informações na Wikipédia: Lista de episódios de Naruto Shippuden

Por Beatriz Milz em [Tutoriais](#)

LEIA MAIS SOBRE

[Análises](#) [Boas práticas](#) [Conceitos](#) [Debugging](#) [Discussões](#) [Divulgação](#) [Erros](#) [Pacotes](#) [R](#) [Top 10](#) [Tutoriais](#)

