

# O QUE É O SHINY?

Allan Clemente\*

## RESUMO

Interagir, analisar e comunicar são três pilares indispensável para se contar boas história com dados em artigos ou dashboards profissionais. Ainda hoje grande parte das análises são estáticas e oferecem apenas a visão do pesquisador acerca dos dados analisados. Nesse contexto, o usuário tem o seu direito a uma nova opinião sobre o mesmo conjunto de dados limitado ou cerceado. Justamente para mudar essa realidade o Shiny foi desenvolvido. Em linhas gerais, essa ferramenta possibilita que análise de dados científicos atinja um novo patamar, adequando-se ao paradigma da programação reativa.

Palavras-chave: programação reativa. Dashboards. Data Storytelling.

## 1. INTRODUÇÃO: O QUE É SHINY?

Shiny é um pacote do R que facilita o desenvolvimento de aplicativos web. Essa ferramenta tornar rápido e simples o compartilhamento, a análises e a apresentação de gráficos usando R. Além disso, permite que usuários possam interagir com esses elementos, personalizando sua consulta ou coleta de informação (Beeley, 2018).

Com essa ferramenta é possível hospedar aplicativos independentes na web, incorporá-los em documentos R Markdown ou construir painéis ou dashboards. Além disso, é possível estender as funcionalidades do Shiny com temas CSS , html widgets e JavaScript (Chang et al., 2021). Alguns exemplos podem ser visto em “*Artigo1.R*” clicando [aqui](#).

### 1.1 O que é programação reativa?

A programação reativa é baseada principalmente nos conceitos: (i) atualização de variáveis no tempo (também conhecidos como sinais ou comportamentos), (ii) fluxos de eventos discretos que atualizam as aplicações, (iii) rastreamento automático de dependências e (iv) propagação automatizada de mudanças (Salvaneschi et al., 2015).

Muitos sistemas de software modernos são reativos: eles respondem à ocorrência de eventos de interesse realizando alguns cálculos, que podem, por sua vez, desencadear novos eventos. Os exemplos variam de interfaces gráficas de usuário, que reagem à entrada dos

---

\* [allan.clemente@alu.ufc.br](mailto:allan.clemente@alu.ufc.br) – Doutorando em Saneamento Ambiental – DEHA – UFC – [www.ds4all.com.br](http://www.ds4all.com.br)

usuários, a sistemas embarcados, que reagem aos sinais vindos do hardware, a aplicações de monitoramento e controle, que reagem às mudanças no ambiente externo (Mosteo, 2020).

## 1.2 Porque pesquisadores devem pensar nisso?

A visualização da informação tem sido amplamente utilizada como um meio eficaz de transmitir informação, ao invés de simplesmente apresentar dados. Nesse contexto, para atender essa demanda e considerando a especificidade dos diversos usuários finais, várias ferramentas de análise e visualização de dados surgiram com considerações diferentes (Mei et al., 2018).

Paralelamente, projetar, implementar e manter um software reativo é indiscutivelmente difícil, sendo necessários os esforços combinados de profissionais e pesquisadores de design, programação e engenharia de software para que esses sistemas sejam eficazes (Salvaneschi et al., 2015).

O Shiny vai na contramão dessa realidade. Essa ferramenta permite que pesquisadores de qualquer área, sem nenhuma habilidade de desenvolvimento web por exemplo, possam implementar suas primeiras soluções reativas usando linguem R, amplamente conhecida na comunidade científica (Chang et al., 2021).

Portanto, além dos gráficos e análises sugeridas pelos autores dos artigos, a adoção de aplicações reativas para a comunidade científica pode estender a experiência do usuário, possibilitando a descoberta de novas facetas (não observados pelos autores originais), ampliando a chance de citação e contribuindo para o avanço científicos.

## 2. PRIMEIROS PASSOS COM O SHINY

Após entender o panorama geral da proposta do Shiny, é hora de implementar no R o primeiro aplicativo. Mas antes disso, iniciaremos com visualizando alguns exemplos ilustrativos do poder dessa biblioteca. Para as versões mais modernas do RStudio, o Shiny já vem pré-instalado. Todavia, caso use outra IDE basta instalar o pacote usando `install.packages("shiny")`.

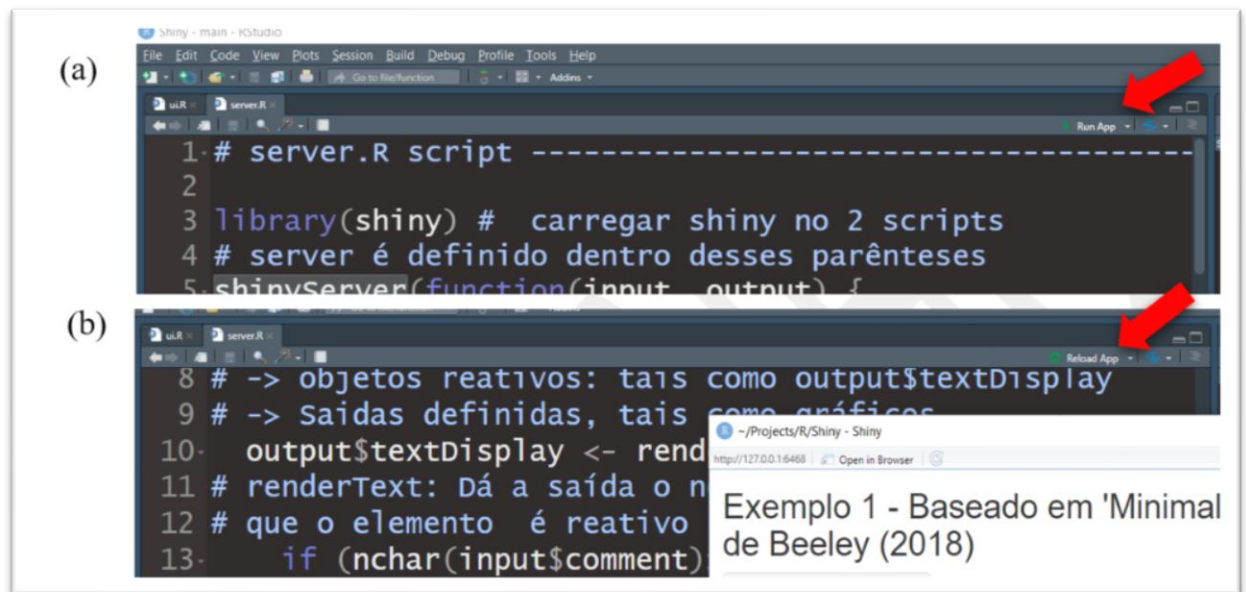
### 2.1 Estrutura do programa

Alguns autores como Beeley (2018) afirmam que aplicações com os Shiny são mais fáceis de se construir usando dois scripts: “*server.R*” e “*ui.R*”. Esses scripts são mantidos na mesma pasta e devem sempre ser nomeados dessa forma. Um exemplo pode ser visto [aqui](#).

Para executar um programa Shiny em uma máquina local, deve-se:

- Manter os scripts `server.R` e `ui.R` na mesma pasta;
- Tornar o caminho da pasta o diretório de trabalho do R (usando o comando `setwd()`, por exemplo);
- Carregar o shiny, usando `library(shiny)`;
- Digitar `runApp()` no console ou seguir as instruções da Figura 1.

Figura 1 – Executando uma aplicação Shiny usando o RStudio. Em (a) A Seta vermelha indica o botão “Run App” que deve ser clicado para rodar a aplicação. Ressalta-se em (b) que após iniciar a aplicação, as atualizações também podem ser implementadas clicando no mesmo botão que apresenta um título diferente (*Reload App*). É imprerterível cumprir todas as etapas anteriores.



É importante salientar que existem inúmeras maneiras de criar e executar uma aplicação com o Shiny. Os procedimentos apresentados aqui ilustram apenas algumas delas e nos próximos capítulos serão apresentadas outras.

### 2.1.1 O script `ui.R`

Nesse script encontram-se as configurações da interface do usuário ou UI (user interface). Sendo assim, é nessa parte que o usuário consegue interagir com a aplicação. Por causa disso a `ui.R` deve ser cuidadosamente desenvolvida. O arquivo `ui.R` pode ser desenvolvido de várias formas, aqui é apresentada uma.

Para ilustrar algumas partes importantes de um script `ui.R`, cita-se:

- `shinyUI(pageWithSidebar(...))`: informa que layout vanilla UI está sendo usado. Nesse layout todos os controles estão no lado esquerdo, oferecendo um grande espaço no lado direito para incluir gráficos, tabelas e texto. Todos os elementos da IU são definidos dentro desta instrução;
- `headerPanel()`: dá um título ao aplicativo;

- `sidebarPanel()`: configura os controles do aplicativo e geralmente contém todos os widgets de entrada;
- `widgets`: é um componente incorporado que é utilizado para simplificar o acesso ou a coleta de informações em uma aplicação web. São exemplos:
  - `textInput()`: widget simples que coleta texto de uma caixa de texto com a qual os usuários podem interagir usando o teclado;
  - `sliderInput()`: constrói um widget de controle deslizante (slider) para selecionar um valor numérico em um intervalo pré-definido;
  - Para outros elementos - `actionButton()`, `checkboxGroupInput()`, `checkboxInput()`, `dateInput()`, `dateRangeInput()`, `fileInput()`, `numericInput()`, `passwordInput()`, `radioButtons()`, `selectInput()`, `submitButton()`, `textAreaInput()` e `varSelectInput()` – consultar o help e o item 2.1.3.
- `mainPanel()`: configura a área de saída e é a área para lidar com texto reativo gerado dentro do servidor.

O arquivo `ui.R` pode ser desenvolvido de várias formas, [aqui](#) é apresentado uma.

### 2.1.2 O script `server.R`

No geral, o script `server.R` contém duas partes importantes: os objetos reativos e as saídas. Para ilustrar algumas partes desse script, pode-se citar:

- `shinyServer(...{...})`: define que parte do Shiny que vai lidar com os objetos reativos e as saídas;
- `output$`Saída_definida_na_ui.R` <- renderText({..})`: Define um elemento de saída. Nessa instrução pelo menos três detalhes são importantes:
  - Dá a saída um nome ( ao invés de “`Saída_definida_na_ui.R`” deve-se usar o nome definido na `ui.R` ) para que possa ser referenciado em `ui.R` (geralmente na última parte de `ui.R` );
  - Diz ao Shiny que o conteúdo contido nele é reativo, ou seja, é atualizado quando suas entradas forem alteradas) e que assume, no caso de `renderText`, a forma de texto;
  - Para gráficos usar `renderPlot({..})` que renderiza um gráfico reativo. Para renderizar outros elementos pesquisar outras funções no help;

- Para o processamento real, as entradas são lidas a partir da ui.R usando `input$Nome_definido_na_ui.R'`;

Para verificar um exemplo comentado do script server.R clique [aqui](#)

### 2.1.3 Tipos de widget

Nessa seção serão apresentados alguns dos principais widgets. Beeley (2018) desenvolveu um aplicativo Shiny que mostrar os principais widgets em ação. Para executá-lo, basta inserir o seguinte comando: `runGist(6571951)`. Para maiores detalhes acessar [aqui](#).

Continua em breve...

## REFERÊNCIA

- Beeley, C., 2018. Web Application Development with R using Shiny, Surveillance and Society. Packt Publishing Ltd.
- Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson, J., Dipert, A., Borges, B., 2021. Shiny [WWW Document]. URL <https://shiny.rstudio.com/> (accessed 6.12.21).
- Mei, H., Ma, Y., Wei, Y., Chen, W., 2018. The design space of construction tools for information visualization: A survey. J. Vis. Lang. Comput. 44, 120–132. <https://doi.org/10.1016/j.jvlc.2017.10.001>
- Mosteo, A.R., 2020. Reactive programming in Ada 2012 with RxAda. J. Syst. Archit. 110, 101784. <https://doi.org/10.1016/j.sysarc.2020.101784>
- Salvaneschi, G., Margara, A., Tamburrelli, G., 2015. Reactive Programming: A Walkthrough. Proc. - Int. Conf. Softw. Eng. 2, 953–954. <https://doi.org/10.1109/ICSE.2015.303>