



# Desenvolvimento de aplicativo em Shiny para Visualização de Dados Meteorológicos no Rio Grande do Sul

Pedro Dresseno John, Yuri Dessimon, Márcia Barbian

21 de setembro de 2025

## O que é um shiny?

- Pacote de *R* que permite fazer sites dinâmicos com código de *R*

## O que é um shiny?

- Pacote de *R* que permite fazer sites dinâmicos com código de *R*

# Porque usar shiny?

- É bonito ☺

## Porque usar shiny?

- É bonito ☺
- Permite mostrar as relações entre muitas variáveis

# Porque usar shiny?

- É bonito ☺
- Permite mostrar as relações entre muitas variáveis
- Facilmente *hosteável* na internet

## Porque usar shiny?

- É bonito ☺
- Permite mostrar as relações entre muitas variáveis
- Facilmente *hosteável* na internet
- Impressiona muito as pessoas (principalmente professores & colegas)

## Estrutura básica de um shiny

- *ui* - user interface (Interface de Usuário)

## Estrutura básica de um shiny

- *ui* - user interface (Interface de Usuário)
- *server* (Servidor)

## Estrutura básica de um shiny

- *ui* - user interface (Interface de Usuário)
- server (Servidor)

*ui*  $\Rightarrow$  *server*  $\Rightarrow$  *ui*

# Estrutura básica de um shiny

- *ui* - user interface (Interface de Usuário)
- server (Servidor)

*ui*  $\Rightarrow$  *server*  $\Rightarrow$  *ui*

## Passo 1: *ui*

Recebe os valores que o usuário coloca nos botões e os coloca em um objeto chamado *input* que é enviado ao servidor

# Estrutura básica de um shiny

- *ui* - user interface (Interface de Usuário)
- server (Servidor)

*ui*  $\Rightarrow$  *server*  $\Rightarrow$  *ui*

## Passo 1: *ui*

Recebe os valores que o usuário coloca nos botões e os coloca em um objeto chamado *input* que é enviado ao servidor

## Passo 2: *server*

Recebe o objeto *input*, e renderiza os gráficos, os colocando em um objeto chamado *output*, que é enviado a interface de usuário novamente

# Estrutura básica de um shiny

- *ui* - user interface (Interface de Usuário)
- server (Servidor)

*ui*  $\Rightarrow$  *server*  $\Rightarrow$  *ui*

## Passo 1: *ui*

Recebe os valores que o usuário coloca nos botões e os coloca em um objeto chamado *input* que é enviado ao servidor

## Passo 2: *server*

Recebe o objeto *input*, e renderiza os gráficos, os colocando em um objeto chamado *output*, que é enviado a interface de usuário novamente

## Passo 3: *ui*

Recebe o objeto final de *output* e mostra o que o servidor renderizou ao usuário

## Dados INMET - A maneira difícil

- Os dados originalmente foram retirados manualmente das tabelas listadas no site do INMET: Dados das Estações

## Dados INMET - A maneira fácil

- O INMET tem um acervo histórico que permite baixar dados agregados anualmente de todas as estações simultaneamente: Dados Históricos
- O banco também vem já com todas as estações agregadas

## Variáveis do Banco

As observações são automaticamente feitas de hora em hora, normalmente com três versões: **Ins**, **Max** e **Min**

## Variáveis do Banco

As observações são automaticamente feitas de hora em hora, normalmente com três versões: **Ins, Max e Min**

- Temperatura (Cº)
  - Ins, Max e Min

## Variáveis do Banco

As observações são automaticamente feitas de hora em hora, normalmente com três versões: **Ins, Max e Min**

- Temperatura (Cº)
  - Ins, Max e Min
- Umidade (%)
  - Ins, Max e Min

## Variáveis do Banco

As observações são automaticamente feitas de hora em hora, normalmente com três versões: **Ins, Max e Min**

- Temperatura (C°)
  - Ins, Max e Min
- Umidade (%)
  - Ins, Max e Min
- Ponto do Orvalho (C°)
  - Ins, Max e Min

## Variáveis do Banco

As observações são automaticamente feitas de hora em hora, normalmente com três versões: **Ins, Max e Min**

- Temperatura (C°)
  - Ins, Max e Min
- Umidade (%)
  - Ins, Max e Min
- Ponto do Orvalho (C°)
  - Ins, Max e Min
- Pressão (hPa)
  - Ins, Max e Min

## Variáveis do Banco

As observações são automaticamente feitas de hora em hora, normalmente com três versões: **Ins, Max e Min**

- Temperatura (C°)
  - Ins, Max e Min
- Umidade (%)
  - Ins, Max e Min
- Ponto do Orvalho (C°)
  - Ins, Max e Min
- Pressão (hPa)
  - Ins, Max e Min
- **Vento** (m/s)
  - Velocidade, Direção e Rajada

## Variáveis do Banco

As observações são automaticamente feitas de hora em hora, normalmente com três versões: **Ins, Max e Min**

- Temperatura (C°)
  - Ins, Max e Min
- Umidade (%)
  - Ins, Max e Min
- Ponto do Orvalho (C°)
  - Ins, Max e Min
- Pressão (hPa)
  - Ins, Max e Min
- **Vento** (m/s)
  - Velocidade, Direção e Rajada
- **Radiação** (KJ/m)

## Variáveis do Banco

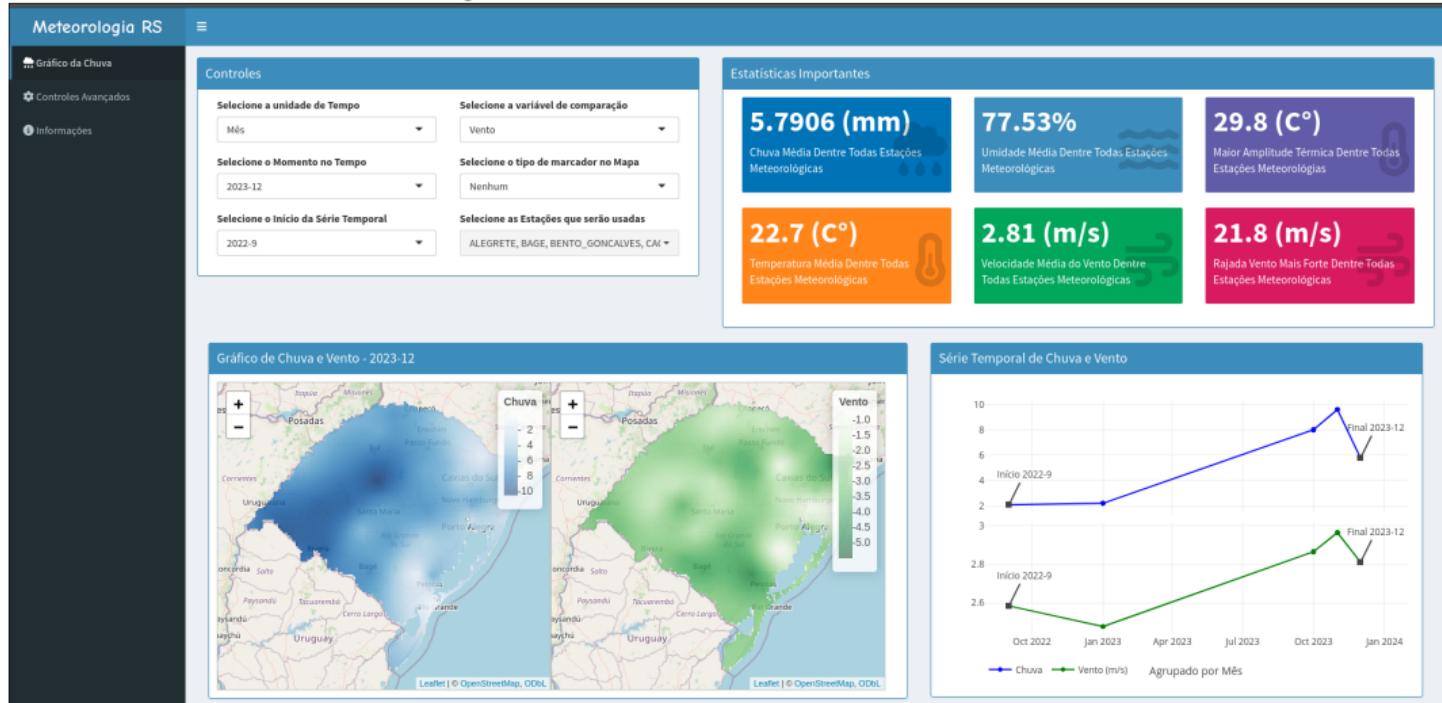
As observações são automaticamente feitas de hora em hora, normalmente com três versões: **Ins**, **Max** e **Min**

- Temperatura (C°)
  - Ins, Max e Min
- Umidade (%)
  - Ins, Max e Min
- Ponto do Orvalho (C°)
  - Ins, Max e Min
- Pressão (hPa)
  - Ins, Max e Min
- **Vento** (m/s)
  - Velocidade, Direção e Rajada
- **Radiação** (KJ/m)
- **Chuva** (mm)

# Página inicial do Aplicativo

Pacotes usados na ui: shiny, shinyWidgets, shinydashboard, leaflet e plotly

Pacotes extras usados no servidor: dplyr, htmltools, sp, sf, gstat, ggplot2, terra, leaflet.minicharts e leafsync



## Tratamento do Banco - Antes do shiny

Para que o banco fosse ideal para trabalhar no *shiny* os seguintes passos foram feitos:

## Tratamento do Banco - Antes do shiny

Para que o banco fosse ideal para trabalhar no *shiny* os seguintes passos foram feitos:

1. Agrupar os dados que são observados de hora em hora, para criar quatro variáveis de data:
  - Dia
  - Semana
  - Mês
  - Ano

## Tratamento do Banco - Antes do shiny

Para que o banco fosse ideal para trabalhar no *shiny* os seguintes passos foram feitos:

1. Agrupar os dados que são observados de hora em hora, para criar quatro variáveis de data:
  - Dia
  - Semana
  - Mês
  - Ano
2. Manualmente inserir a latitude e longitude baseada no nome da estação

# Tratamento do Banco - Antes do shiny

Para que o banco fosse ideal para trabalhar no *shiny* os seguintes passos foram feitos:

1. Agrupar os dados que são observados de hora em hora, para criar quatro variáveis de data:
  - Dia
  - Semana
  - Mês
  - Ano
2. Manualmente inserir a latitude e longitude baseada no nome da estação
3. Extrair o código da estação

dplyr & shiny & func. base

list.files()  
file.path()  
%in%  
switch()  
getElement()  
get()  
dplyr::rename()  
dplyr::select()  
dplyr::group\_by()  
dplyr::summarise()  
shiny::reactive()

## Shapefile e geobr

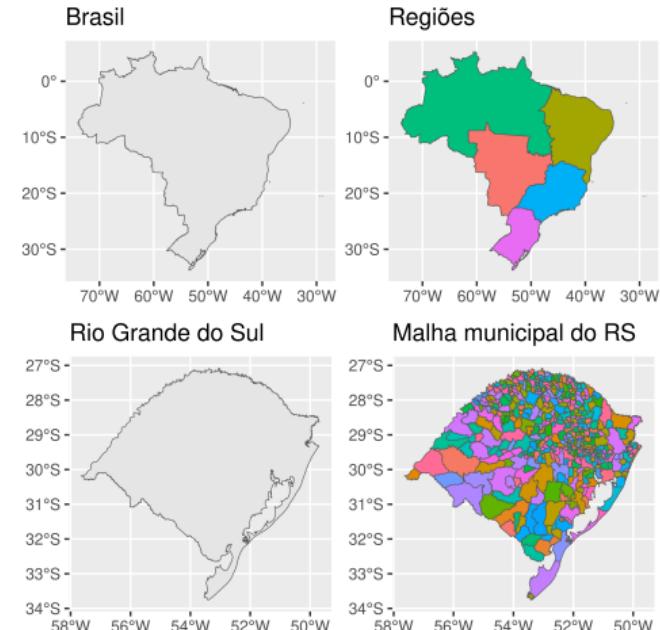
- Arquivo que nos dá o formato de uma região, seja estado, país, município, cidade, etc.
- No *R* temos o pacote *geobr* que permite carregar esses arquivos

# Shapefile e geobr

- Arquivo que nos dá o formato de uma região, seja estado, país, município, cidade, etc.
- No R temos o pacote *geobr* que permite carregar esses arquivos

## Código

```
br = geobr::read_country()  
regioes = geobr::read_region()  
rs = geobr::read_state("RS")  
mun = geobr::read_municipality("RS")
```



## Controles do Aplicativo

Pode-se selecionar:

- A unidade de tempo sendo usada

## Controles do Aplicativo

Pode-se selecionar:

- A unidade de tempo sendo usada
- O momento no tempo a partir do qual a krigagem será feita

## Controles do Aplicativo

Pode-se selecionar:

- A unidade de tempo sendo usada
- O momento no tempo a partir do qual a krigagem será feita
- Uma data anterior para montagem de um gráfico de séries temporais

## Controles do Aplicativo

Pode-se selecionar:

- A unidade de tempo sendo usada
- O momento no tempo a partir do qual a krigagem será feita
- Uma data anterior para montagem de um gráfico de séries temporais
- Uma variável para a krigagem secundária (entre **Vento**, **Temperatura** ou **Umidade**)

## Controles do Aplicativo

Pode-se selecionar:

- A unidade de tempo sendo usada
- O momento no tempo a partir do qual a krigagem será feita
- Uma data anterior para montagem de um gráfico de séries temporais
- Uma variável para a krigagem secundária (entre **Vento**, **Temperatura** ou **Umidade**)
- A presença e tipo de marcador no mapa para as estações meteorológicas

# Controles do Aplicativo

Pode-se selecionar:

- A unidade de tempo sendo usada
- O momento no tempo a partir do qual a krigagem será feita
- Uma data anterior para montagem de um gráfico de séries temporais
- Uma variável para a krigagem secundária (entre **Vento**, **Temperatura** ou **Umidade**)
- A presença e tipo de marcador no mapa para as estações meteorológicas
- Quais estações serão usadas ou excluídas

Controles

Selecione a unidade de Tempo	Selecione a variável de comparação
<input type="button" value="Dia"/>	<input type="button" value="Vento"/>
Selecione o Momento no Tempo	Selecione o tipo de marcador no Mapa
<input type="button" value="2023-12-31"/>	<input type="button" value="Nenhum"/>
Selecione o Início da Série Temporal	Selecione as Estações que serão usadas
<input type="button" value="2023-12-24"/>	<input 545="" 629="" 640="" 685"="" data-label="Section-Header" type="button" value="ALEGRETE, BAGE, BENTO_GONCALVES, CA(&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;/div&gt;&lt;div data-bbox="/> <h2>Código</h2>

```
shiny::selectInput()  
shinyWidgets::pickerInput()
```

## Value Boxes

Caixas coloridas com imagens e valores dinâmicos, muito bonitas. Presentes em vários pacotes, incluindo o `shinydashboard`.



## Gráfico de séries temporais

- ui reativa pro começo da série temporal
  - uiOutput
  - renderUI

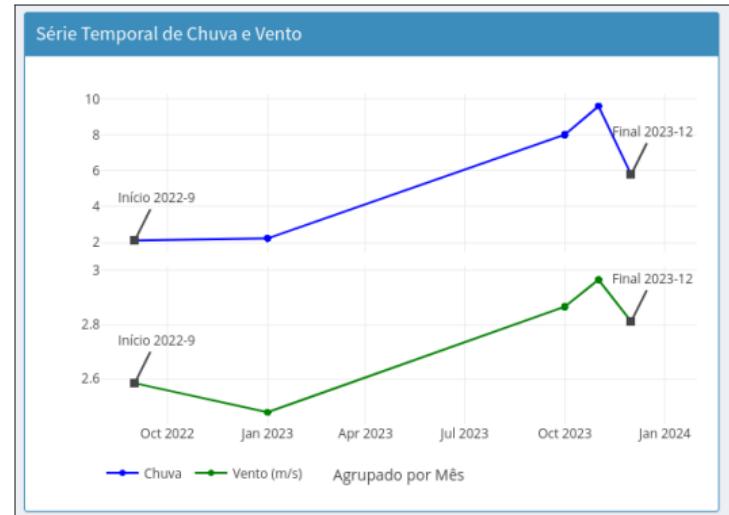
# Gráfico de séries temporais

- ui reativa pro começo da série temporal
  - uiOutput
  - renderUI
- *plotly* ótimo pacote de gráficos

## Código

```
plotly::plot_ly()  
plotly::add_trace()  
plotly::layout()  
plotly::layout(annotations = ...)  
plotly::subplot()
```

```
plotly::ggplotly()
```

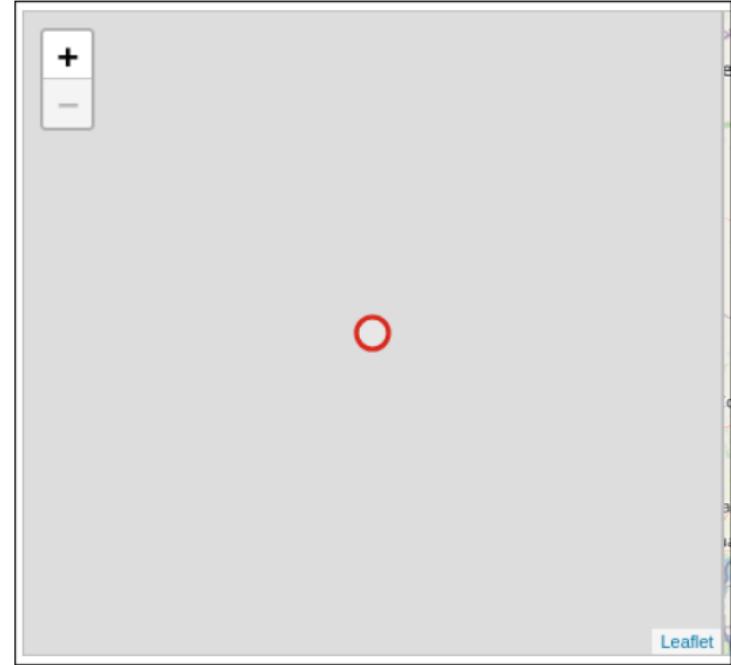


# leaflet

Montando um gráfico do *leaflet*

## Código

```
|>  
leaflet::leaflet()
```

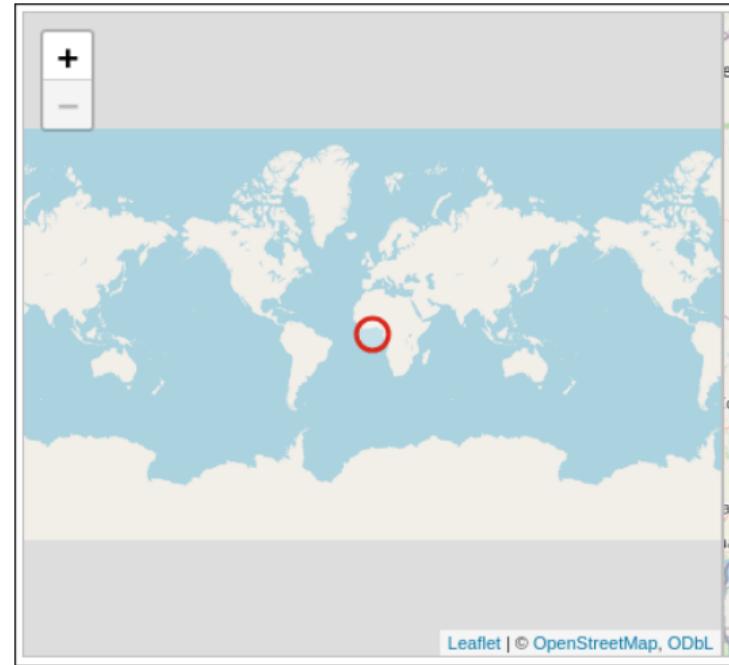


# leaflet

Montando um gráfico do *leaflet*

## Código

```
|>  
leaflet::leaflet()  
leaflet::addTiles()
```

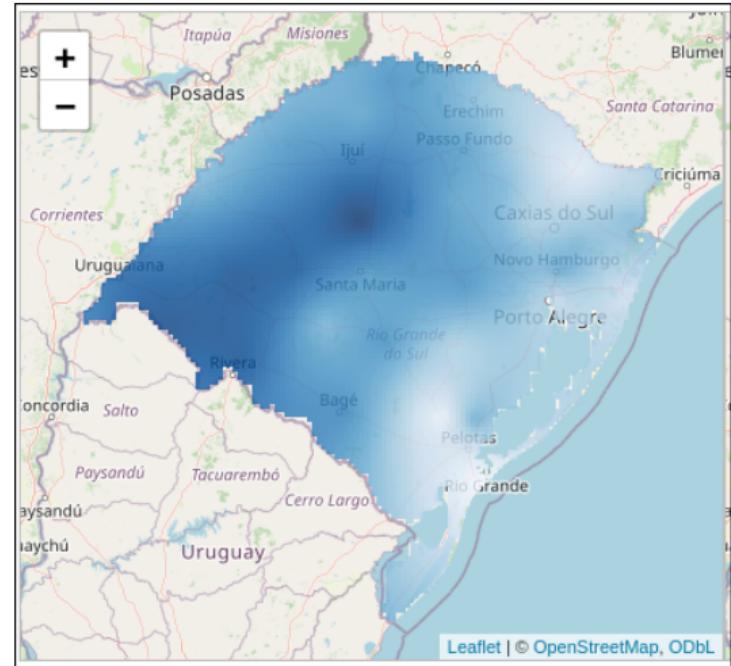


# leaflet

Montando um gráfico do *leaflet*

## Código

```
|>  
leaflet::leaflet()  
leaflet::addTiles()  
leaflet::addRasterImage()
```

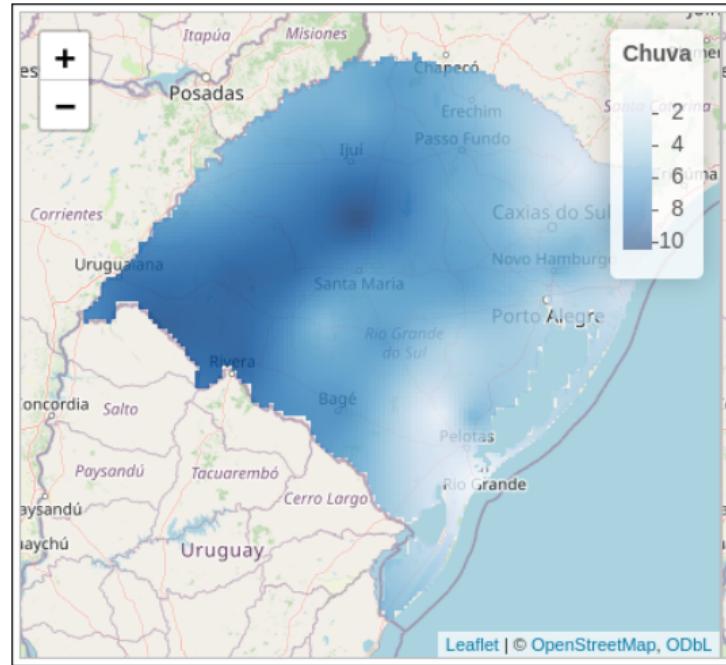


# leaflet

Montando um gráfico do *leaflet*

## Código

```
|>  
leaflet::leaflet()  
leaflet::addTiles()  
leaflet::addRasterImage()  
leaflet::addLegend()
```

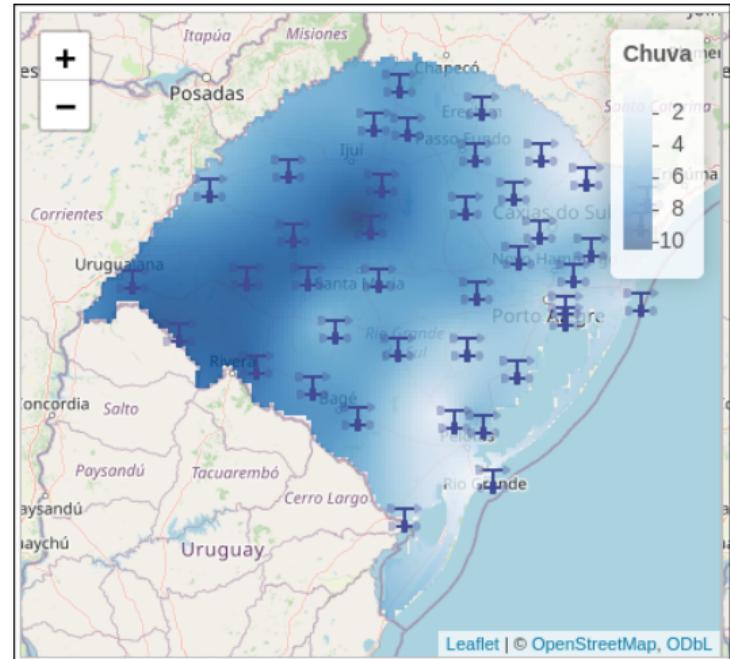


# leaflet

Montando um gráfico do *leaflet*

## Código

```
|>  
leaflet::leaflet()  
leaflet::addTiles()  
leaflet::addRasterImage()  
leaflet::addLegend()  
leaflet::addMarkers()
```

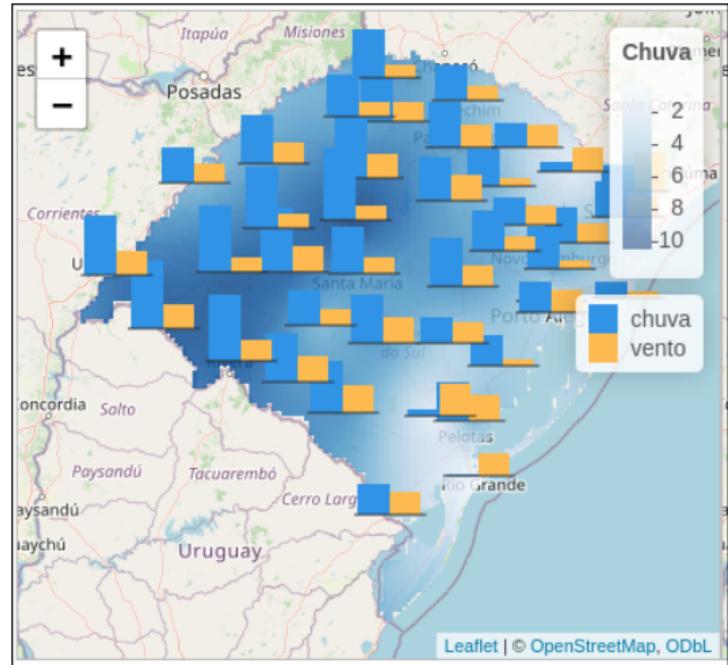


# leaflet

Montando um gráfico do *leaflet*

## Código

```
|>  
leaflet::leaflet()  
leaflet::addTiles()  
leaflet::addRasterImage()  
leaflet::addLegend()  
leaflet::addMarkers()  
leaflet.minicharts::addMinicharts()
```



# leaflet

Montando um gráfico do *leaflet*

## Código

```
|>
leaflet::leaflet()
leaflet::addTiles()
leaflet::addRasterImage()
leaflet::addLegend()
leaflet::addMarkers()
leaflet.minicharts::addMinicharts()

leaflet::setMaxBounds()
```

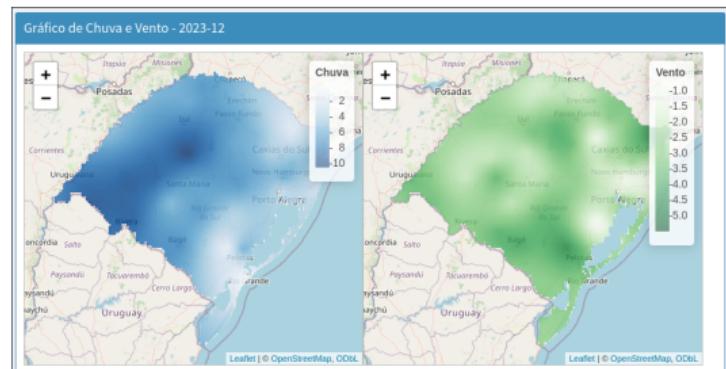
# leafsync

Útil para colocar dois ou mais gráficos do *leaflet* juntos e também bem fácil de usar:

## Código leafsync

```
mapa_1 = leaflet::leaflet() |> ...  
mapa_2 = leaflet::leaflet() |> ...
```

```
leafsync::sync(mapa_1, mapa_2)
```



# GitHub

O código dessa apresentação e do aplicativo *shiny* estão no GitHub

<https://github.com/-johnpd4/apresentacao-rday5-2025>



## Agradecimentos

Agradecemos à Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) pelo apoio financeiro, por meio do Projeto nº 24/2551-0002361-5, contemplado no Edital 06/2024 – Programa de Pesquisa e Desenvolvimento Voltado a Desastres Climáticos. Este suporte foi fundamental para a realização e o avanço desta pesquisa.

# Referências I

-  Appelhans, Tim e Kenton Russell (2019). *leafsync: Small Multiples for Leaflet Web Maps*. R package version 0.1.0. URL: <https://github.com/r-spatial/leafsync>.
-  Bachelier, Veronique et al. (2021). *leaflet.minicharts: Mini Charts for Interactive Maps*. R package version 0.6.2. DOI: 10.32614/CRAN.package.leaflet.minicharts. URL: <https://CRAN.R-project.org/package=leaflet.minicharts>.
-  Bivand, Roger S., Edzer Pebesma e Virgilio Gomez-Rubio (2013). *Applied spatial data analysis with R, Second edition*. Springer, NY. URL: <https://asdar-book.org/>.
-  Chang, Winston et al. (2025). *shiny: Web Application Framework for R*. R package version 1.11.1. URL: <https://shiny.posit.co/>.
-  Cheng, Joe, Barret Schloerke et al. (2025). *leaflet: Create Interactive Web Maps with the JavaScript Leaflet Library*. R package version 2.2.3. URL: <https://rstudio.github.io/leaflet/>.

## Referências II

-  Cheng, Joe, Carson Sievert et al. (2024). *htmltools: Tools for HTML*. R package version 0.5.8.1. URL: <https://github.com/rstudio/htmltools>.
-  Gräler, Benedikt, Edzer Pebesma e Gerard Heuvelink (2016). “Spatio-Temporal Interpolation using gstat”. Em: *The R Journal* 8 (1), pp. 204–218. URL: <https://journal.r-project.org/archive/2016/RJ-2016-014/index.html>.
-  Henry, Lionel e Hadley Wickham (2025). *rlang: Functions for Base Types and Core R and Tidyverse Features*. R package version 1.1.6. URL: <https://rlang.r-lib.org>.
-  Hijmans, Robert J. (2025). *terra: Spatial Data Analysis*. R package version 1.8-60. URL: <https://rspatial.org/>.
-  Pebesma, Edzer (2018). “Simple Features for R: Standardized Support for Spatial Vector Data”. Em: *The R Journal* 10.1, pp. 439–446. DOI: <10.32614/RJ-2018-009>. URL: <https://doi.org/10.32614/RJ-2018-009>.

## Referências III

-  — (2025). *sf: Simple Features for R*. R package version 1.0-21. URL: <https://r-spatial.github.io/sf/>.
-  Pebesma, Edzer e Roger Bivand (2023). *Spatial Data Science: With applications in R*. Chapman and Hall/CRC. DOI: 10.1201/9780429459016. URL: <https://r-spatial.org/book/>.
-  — (2025). *sp: Classes and Methods for Spatial Data*. R package version 2.2-0. URL: <https://github.com/edzer/sp/>.
-  Pebesma, Edzer e Benedikt Graeler (2025). *gstat: Spatial and Spatio-Temporal Geostatistical Modelling, Prediction and Simulation*. R package version 2.1-4. URL: <https://github.com/r-spatial/gstat/>.
-  Pebesma, Edzer J. (2004). “Multivariable geostatistics in S: the gstat package”. Em: *Computers & Geosciences* 30, pp. 683–691. URL: <https://doi.org/10.1016/j.cageo.2004.03.012>.
-  Pebesma, Edzer J. e Roger Bivand (nov. de 2005). “Classes and methods for spatial data in R”. Em: *R News* 5.2, pp. 9–13. URL: <https://CRAN.R-project.org/doc/Rnews/>.

## Referências IV

-  Perrier, Victor, Fanny Meyer e David Granjon (2025). *shinyWidgets: Custom Inputs Widgets for Shiny*. R package version 0.9.0. URL: <https://github.com/dreamRs/shinyWidgets>.
-  Sievert, Carson (2020). *Interactive Web-Based Data Visualization with R, plotly, and shiny*. Chapman e Hall/CRC. ISBN: 9781138331457. URL: <https://plotly-r.com>.
-  Sievert, Carson et al. (2025). *plotly: Create Interactive Web Graphics via plotly.js*. R package version 4.11.0. URL: <https://plotly-r.com>.
-  Wickham, Hadley (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN: 978-3-319-24277-4. URL: <https://ggplot2.tidyverse.org>.
-  Wickham, Hadley, Winston Chang et al. (2025). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 3.5.2. URL: <https://ggplot2.tidyverse.org>.

## Referências V

-  Wickham, Hadley, Romain François et al. (2023). *dplyr: A Grammar of Data Manipulation*. R package version 1.1.4. URL: <https://dplyr.tidyverse.org>.

Obrigada!

## Temporary page!

$\text{\LaTeX}$  was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because  $\text{\LaTeX}$  now knows how many pages to expect for this document.