

# Taller introductorio a Shiny – aplicaciones web interactivas en R

Virginia Burguete



Para descargar los materiales del taller:

[https://github.com/vburguete/taller\\_shiny](https://github.com/vburguete/taller_shiny)



- Exportaciones de bienes
- Exportaciones por esfuerzo innovador
- Coficiente exportador
- Concentración de las exportaciones
- Diversificación de las exportaciones
- Acerca de
- Tratamiento de la base
- Listado productos

Por producto Por destino

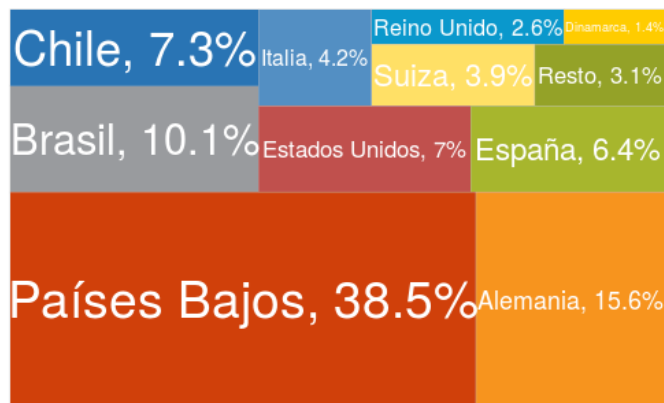
Selector  
datos



### Diversificación de destinos de exportación por producto, según año

Producto: Carne de animales de la especie bovina, fresca o refrigerada.

Año: 2017.



Filtrar:

Pais de destino	Total (USDMM)	Participación (%)
<input type="text" value="All"/>	<input type="text" value="All"/>	<input type="text" value="All"/>
Países Bajos	144,02	38.47%
Alemania	58,51	15.63%
Brasil	37,83	10.10%
Chile	27,43	7.33%
Estados Unidos	26,18	6.99%

1 - 5 (30)

Anterior

Siguiente

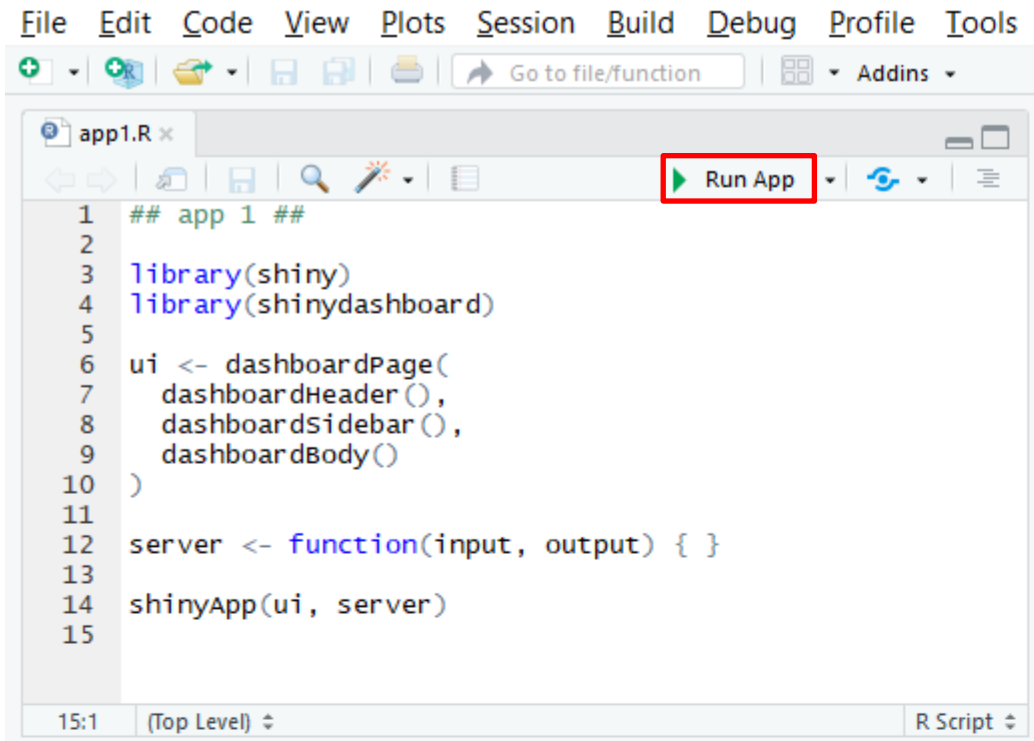
# ¿Qué es?

Shiny es una herramienta para crear aplicaciones web interactivas.

Permite a los usuarios trabajar con los datos sin necesidad de tener conocimientos de programación.

# shinyApp: Componentes

## app 1



```
1 ## app 1 ##
2
3 library(shiny)
4 library(shinydashboard)
5
6 ui <- dashboardPage(
7   dashboardHeader(),
8   dashboardSidebar(),
9   dashboardBody()
10 )
11
12 server <- function(input, output) { }
13
14 shinyApp(ui, server)
15
```

### ui: estructura

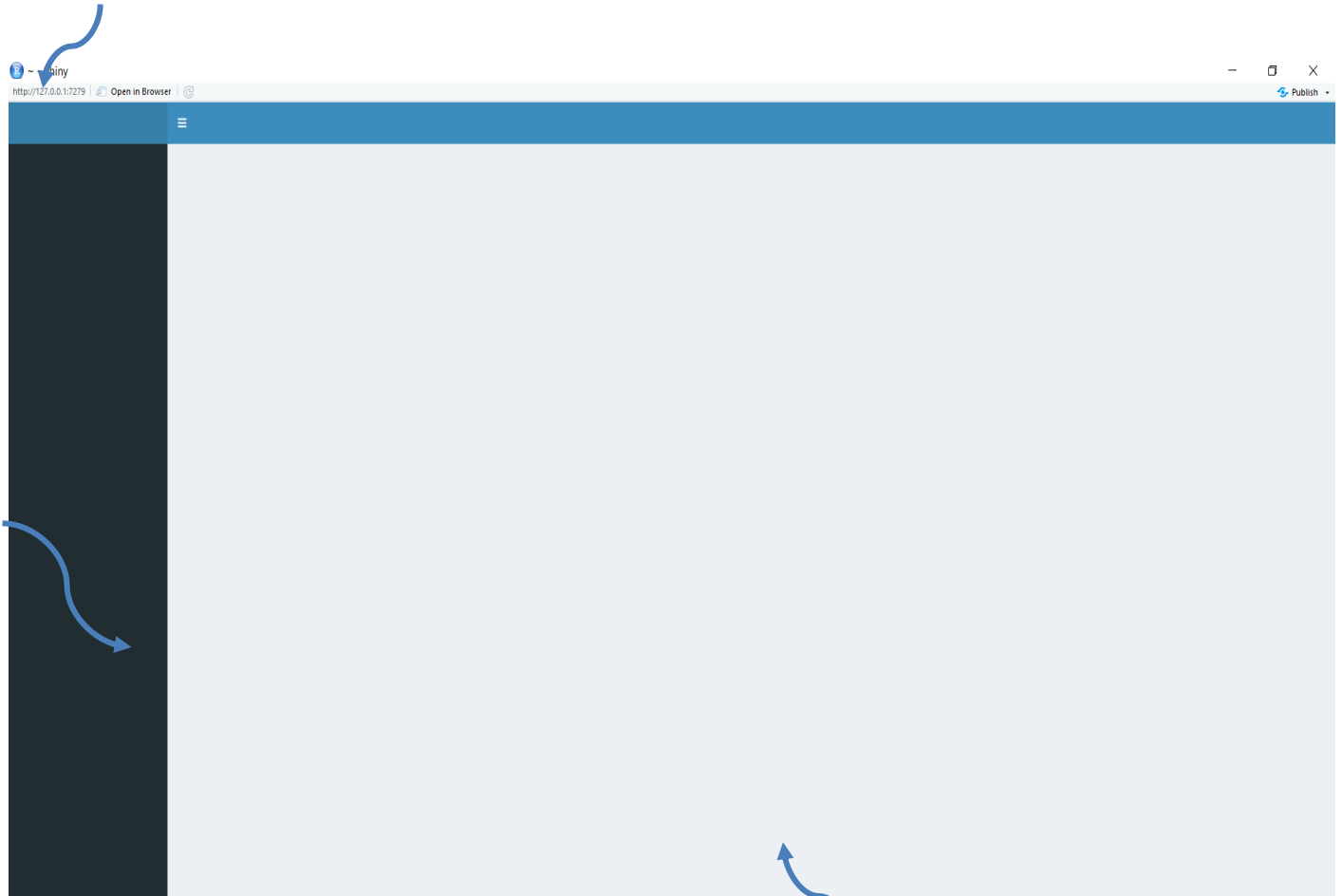
Interfaz de usuario, es donde se programan los elementos con los cuales interactúa el usuario (selectores, botones, cajas, controles, gráficas, etc.)

### server: instrucciones

Server, es donde se programa el funcionamiento de los elementos definidos en la ui (forma y estilo de gráficos, tablas, etc.)

# ui: dashboardPage()

dashboardHeader()



dashboardSidebar()

dashboardBody()

# shinyApp: Argumentos

app 1

```
app1.R x
1
2 ## app 1 ##
3
4 # Descripción:
5 # - definición básica de una app sin argumentos
6
7 # User Interface
8 ui <- dashboardPage(
9   dashboardHeader(),
10  dashboardSidebar(),
11  dashboardBody()
12 )
13
14 # Server
15 server <- function(input, output) { }
16
17 shinyApp(ui, server)
18
```

18:1 (Top Level) R Script



app 2

```
app2.R x
1
2 ## app 2 ##
3
4 # Descripción:
5 # - definición básica de una app
6 # - se incorporan argumentos a cada componente
7
8 # User Interface
9 ui <- dashboardPage(
10
11   dashboardHeader(
12     disable = FALSE,
13     title = NULL,
14     titlewidth = 230
15   ),
16
17   dashboardSidebar(
18     disable = FALSE,
19     width = 230,
20     sidebarMenu(menuItem(""))
21   ),
22
23   dashboardBody(
24     "Idem"
25   )
26 )
27
28 # Server
29 server <- function(input, output) { }
30
31 shinyApp(ui, server)
32
```

19:17 (Top Level) R Script

# dashboardHeader()

app 3

```
app3.R x
1
2 ## app 3 ##
3
4 # Descripción:
5 # - Header: Se agrega título y logo
6 # - Sidebar: Se incorpora pestaña
7 # - Body: Se imprimen textos de distintos tamaños
8
9 # User Interface
10
11 header <- dashboardHeader(
12   title = "Header",
13   titlewidth = 200,
14
15   # Incorpora logo desde la web:
16   tags$li(a(href = 'https://rladies.org/',
17             img(src = "https://rladies.org/wp-content/uploads/2016/12/R-LadiesGlobal.png",
18                 height="30px"),
19             style = "padding-top:10px; padding-bottom:10px;"),
20   class = "dropdown")
21 )
22
23 sidebar <- dashboardSidebar(
24   width = 100,
25   sidebarMenu(
26     menuItem("Sidebar")
27   )
28 )
29
30 body <- dashboardBody(
31   h1("Body"),
32   h2("Body"),
33   h3("Body"),
34   h4("Body"),
35   h5("Body")
36 )
37
38 ui <- dashboardPage(header, sidebar, body)
39
40 # Server
41 server <- function(input, output) { }
42
43 shinyApp(ui, server)
44
45
```



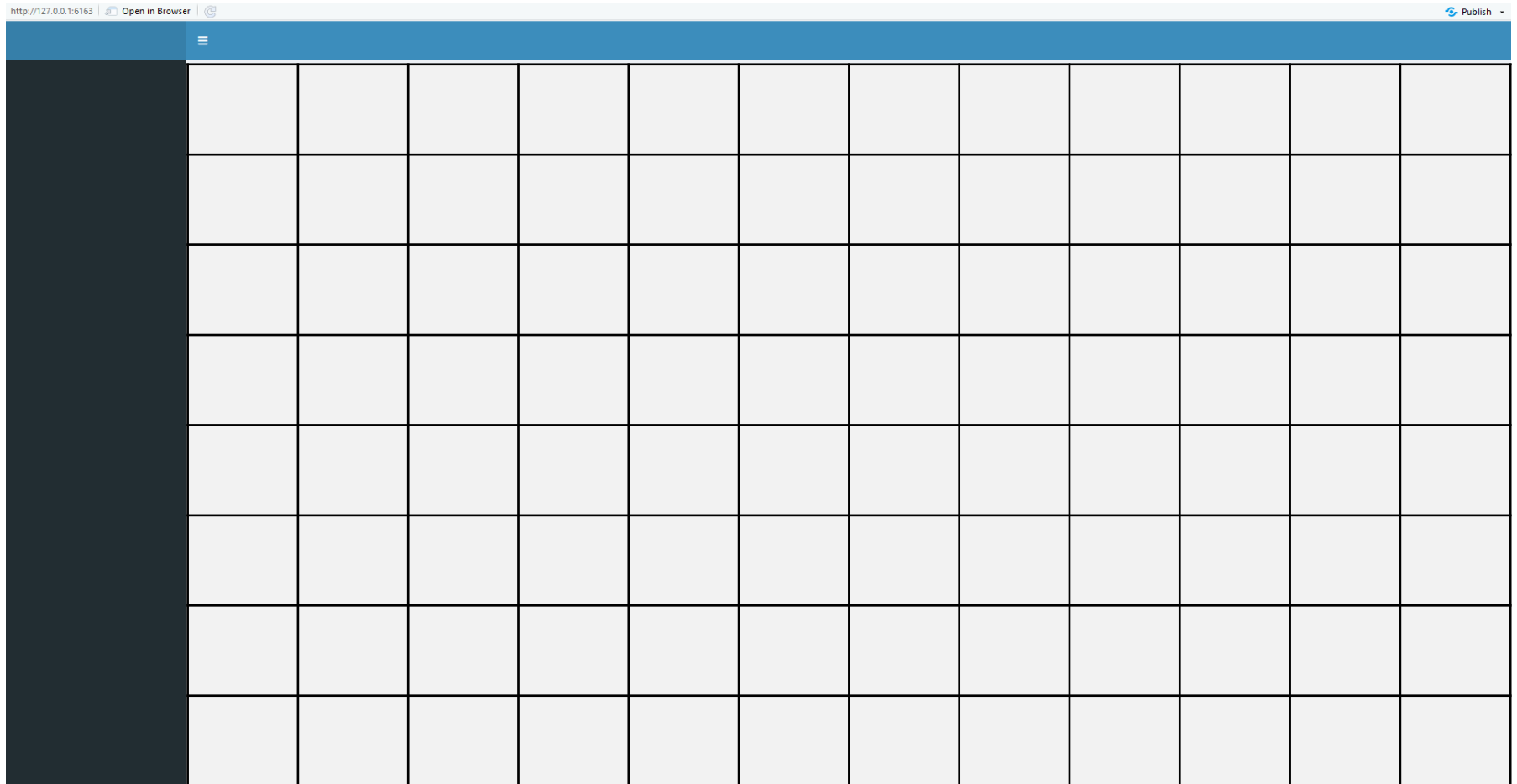
# dashboardSidebar()

app 4

```
app4.R x
1  ## app 4  ##
2
3  # User Interface
4
5  header <- dashboardHeader(
6    title = "Header"
7  )
8
9
10 sidebar <- dashboardSidebar(
11   sidebarMenu(
12     # Se incorporan iconos a cada menuItem (https://glyphsearch.com/)
13     menuItem("Item 1", tabName = "item1",
14              icon = icon("arrow-right", lib = "glyphicon"),          # Glyphicon
15              menuSubItem("SubItem 1", tabName = "subitem1",
16                          icon = icon("arrow-circle-right", lib = "font-awesome")), # Font Awesome
17              startExpanded = TRUE),
18     menuItem("Item 2", tabName = "item2", icon = icon("arrow-right", lib = "glyphicon")),
19     # Se incorpora un slider para seleccionar años
20     sliderInput(inputId = "slider",                                     # ?shiny::sliderInput
21                 label = "",
22                 min = 2010,
23                 max = 2018,
24                 value = 2014,
25                 sep = NULL)
26   )
27 )
28
29 body <- dashboardBody()
30
31 ui <- dashboardPage(header, sidebar, body)
32
33 # Server
34 server <- function(input, output) { }
35
36 shinyApp(ui, server)
37
38
39
40
41
42
43
44
```

44:1 (Top Level) ↕ R Script ↕

# dashboardBody()

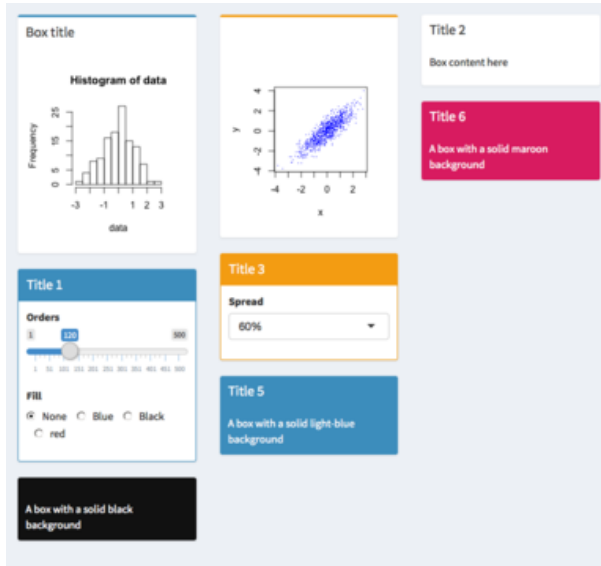



12 columnas

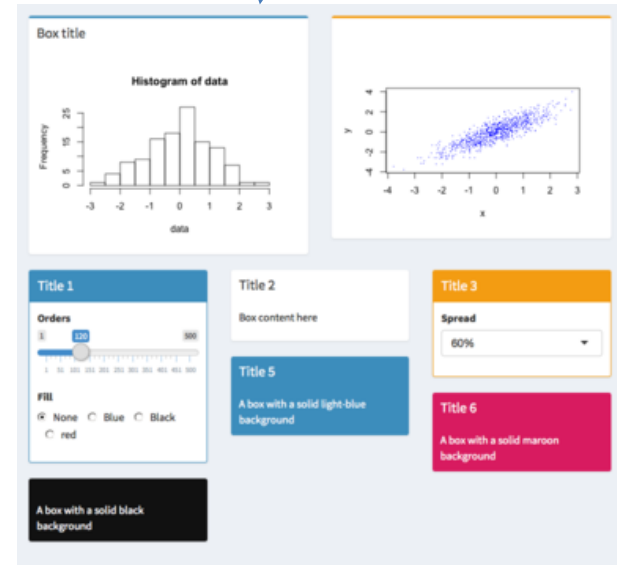


# Diseño: dashboardBody()

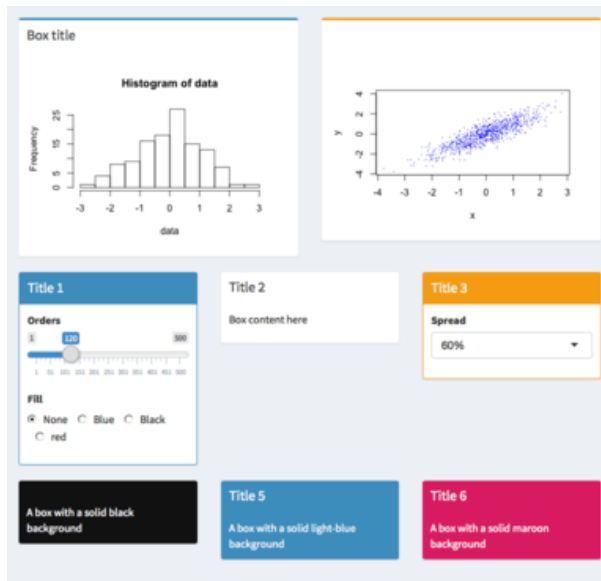
Por columnas



Combinado



Por filas



## app 5

```
app5.R x
Run App

27
28 body <- dashboardBody(
29   tabItems(
30     tabItem(tabName = "subitem1",
31       fluidRow(
32         box(
33           title = "Caja simple con Inputs",
34           width = 4,
35           status = "warning",
36           solidHeader = TRUE,
37           collapsible = TRUE,
38           sliderInput(inputId = "slider",
39             label = "Seleccione año:",
40             min = 2010,
41             max = 2018,
42             value = 2014,
43             sep = NULL)),
44         tabBox(
45           title = "Caja con pestañas",
46           width = 8,
47           tabPanel("Tab1", "Pestaña 1"),
48           tabPanel("Tab2", "Pestaña 2"),
49           selected = "Tab2")),
50     fluidRow(
51       column(4,
52         column(8,
53           infoBox("Info Box", 10000,
54             icon = icon("list"),
55             color = 'aqua',
56             fill = TRUE))
57       ),
58     tabItem(tabName = "item2",
59       fluidRow(
60         box(
61           title = "Caja simple sin fondo",
62           status = "success",
63           textInput("text", "Ingresar texto:"),
64         box(
65           title = "Caja simple con fondo",
66           background = "orange",
67           solidHeader = FALSE)))
68     ))
69
70 ui <- dashboardPage(header, sidebar, body, skin = "black")
71
72 # Server
73 server <- function(input, output) { }
74
75 shinyApp(ui, server)
76
77
```

# ?shinydashboard::box

# ?shinydashboard::validStatuses

# ?shinydashboard::tabBox

# ?shinydashboard::infoBox

# ?shinydashboard::validColors

# ?shiny::textInput

combinado

por filas

67:37 (Top Level) ↕ R Script ↕

# Basic widgets

## Buttons

Action

Submit

## Single checkbox

☒ Choice A

## Checkbox group

☒ Choice 1

☐ Choice 2

☐ Choice 3

## Date input

2014-01-01

## Date range

2017-06-21 to 2017-06-21

## File input

Browse... No file selected

## Help text

Note: help text isn't a true widget, but it provides an easy way to add text to accompany other widgets.

## Numeric input

1

## Radio buttons

☒ Choice 1

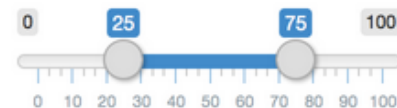
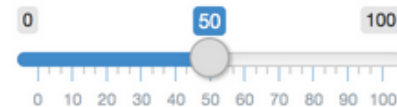
☐ Choice 2

☐ Choice 3

## Select box

Choice 1 ▼

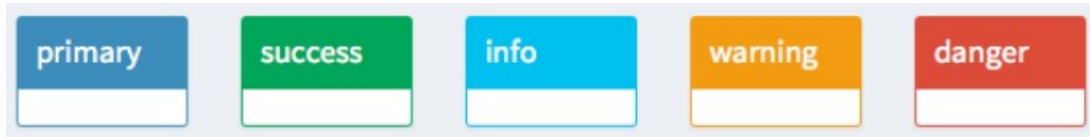
## Sliders



## Text input

Enter text...

# validStatus()



# validColors()



# Reactividad

Cuando una entrada (input) cambia, el servidor reconstruye cada salida (output) que depende de ella.

Inputs

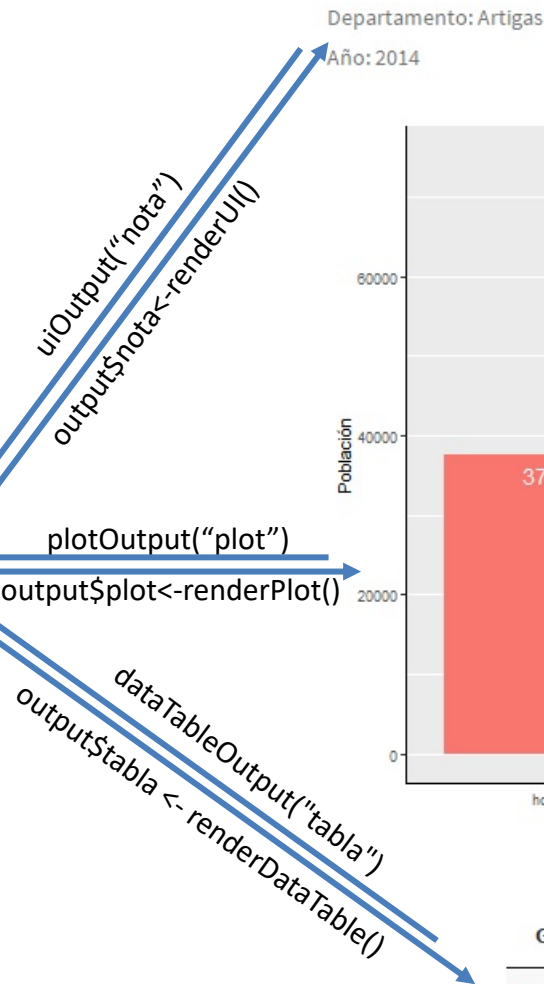
1. Seleccione año:

2011 2014 2018

2011 2012 2013 2014 2015 2016 2017 2018

2. Seleccione departamento:

Artigas

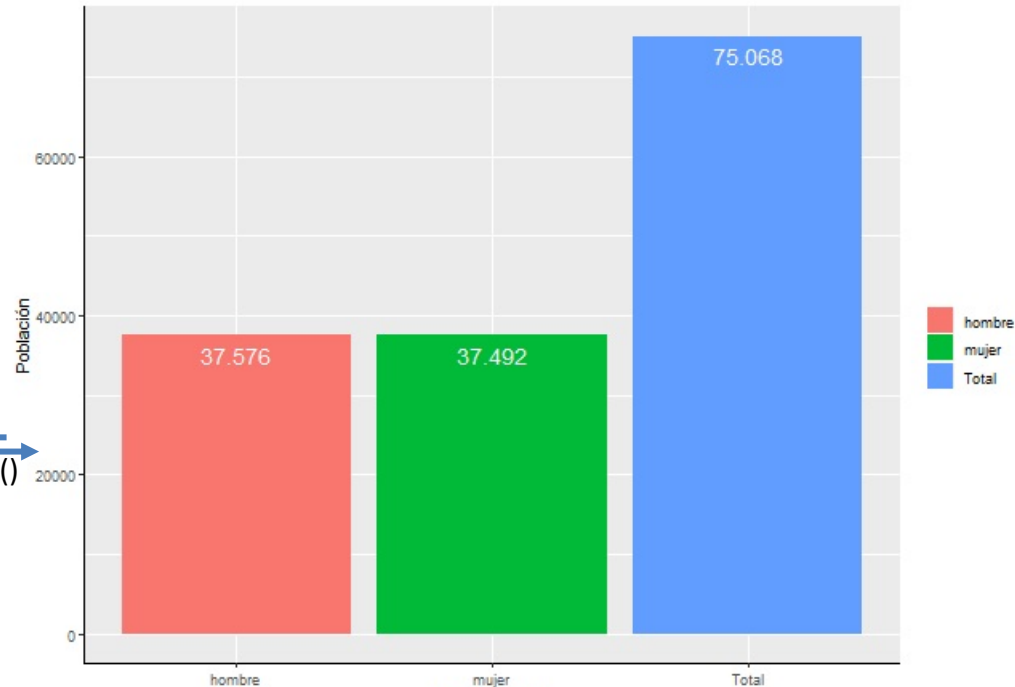


Output

Población por género

Departamento: Artigas

Año: 2014



app 6

Género	Total
hombre	37.576
mujer	37.492
Total	75.068

# Reactividad

Datos extraídos de la  
Encuesta Continua de  
Hogares (2011 a 2018)



Inputs

1. Seleccione año:

2011 2014 2018

2011 2012 2013 2014 2015 2016 2017 2018

2. Seleccione departamento:

Artigas ▼

```
bd
```

```
# A tibble: 977,976 x 7
```

	id	año	ponderador	departamento	genero	edad	edad_categoria
	<chr>	<chr>	<dbl>	<chr>	<chr>	<dbl>	<chr>
1	2011000002001	2011	33	Artigas	hombre	44	30 a 49
2	2011000002002	2011	33	Artigas	mujer	41	30 a 49
3	2011000002003	2011	33	Artigas	hombre	12	menor 14
4	2011000005001	2011	33	Montevideo	hombre	27	14 a 29

```
bd %>% filter(año == "2014",  
              departamento == "Artigas")
```

```
# A tibble: 3,033 x 7
```

	id	año	ponderador	departamento	genero	edad	edad_categoria
	<chr>	<chr>	<dbl>	<chr>	<chr>	<dbl>	<chr>
1	2014000007001	2014	37	Artigas	mujer	53	50 a 64
2	2014000007002	2014	37	Artigas	mujer	28	14 a 29
3	2014000007003	2014	37	Artigas	hombre	24	14 a 29
4	2014000007004	2014	37	Artigas	hombre	16	14 a 29



# app 7

## Inputs

### 1. Seleccione rango de años:



### 2. Seleccione departamento(s):

19 items selected

### 3. Seleccione variable:

edad\_categoria

Tabla

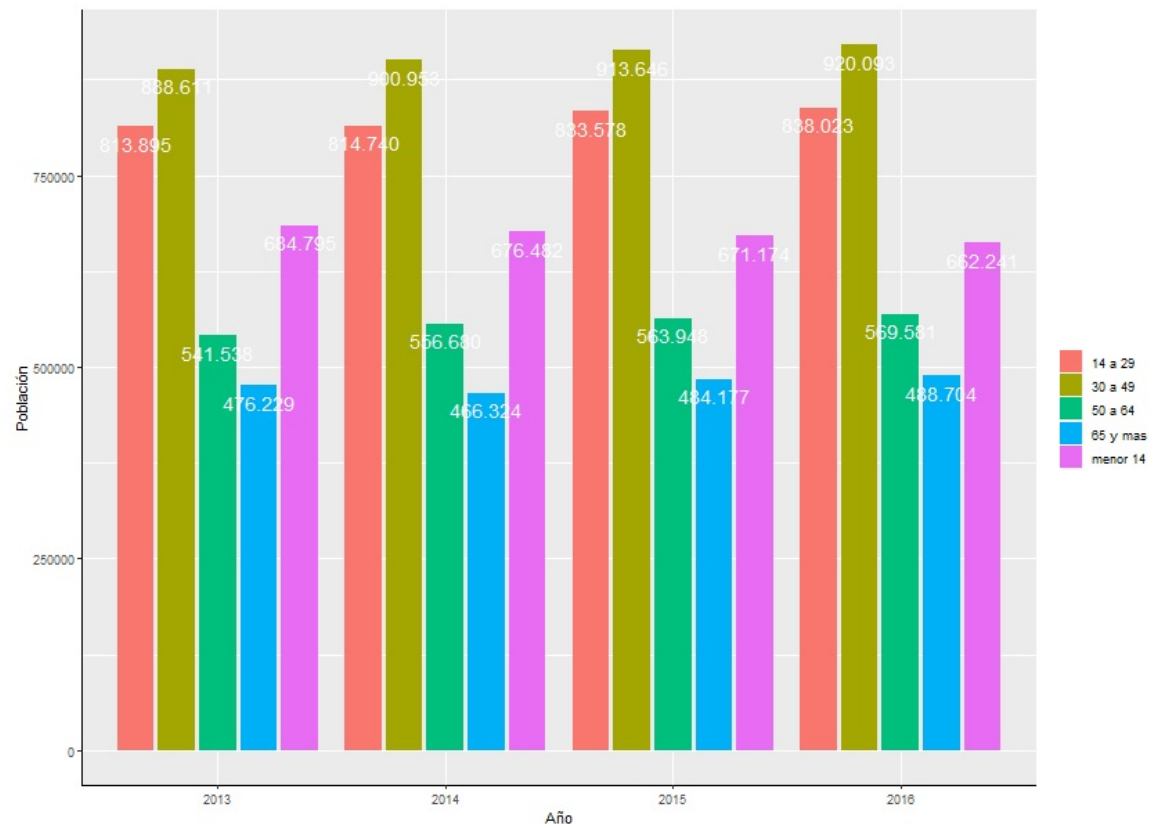
Gráfico

## Población por variable

Variable: edad\_categoria

Total Uruguay

Periodo seleccionado: (2013 - 2016)



# Links de interés

Tutoriales:

<https://rstudio.github.io/shinydashboard>

<https://shiny.rstudio.com/articles/reactivity-overview.html>

Cheatsheets:

<https://shiny.rstudio.com/images/shiny-cheatsheet.pdf>

<https://www.rstudio.com/wp-content/uploads/2015/03/shiny-spanish.pdf>

Íconos:

<https://glyphsearch.com/>

Widgets:

<http://shiny.rstudio.com/gallery/widget-gallery.html>

`shinyWidgets::shinyWidgetsGallery()`

**¡Muchas gracias!**