

Cuadro de mandos en Shiny

I.JOSÉ TUDÓ RAMÍREZ

Qué vamos a ver!



¿Qué es Shiny? - Ventajas - Compartir



Estructura y arquitectura



Pasos a seguir y Resumen



Ejemplo CdM Anulaciones v1.0



~~Preguntas~~

Cerveza Aljibe

¿Qué es Shiny?



- Generación de aplicaciones interactivas.
- Open Source de R desarrollado por R Studio
- `install.packages("shiny")`
- Ejemplos
 - <https://www.rstudio.com/products/shiny/shiny-user-showcase/>
 - <http://www.showmeshiny.com/>

Ventajas



- Poner el foco en la visualización
- Fácil de aprender, >> no necesitas saber sobre HTML, Java script...
- Motivación y distribución de tu proyecto.
- Interacción con el usuario.
- El receptor no tiene que saber R.



Formas de compartir la app



local

- Sesión de R



Shinyapps.io

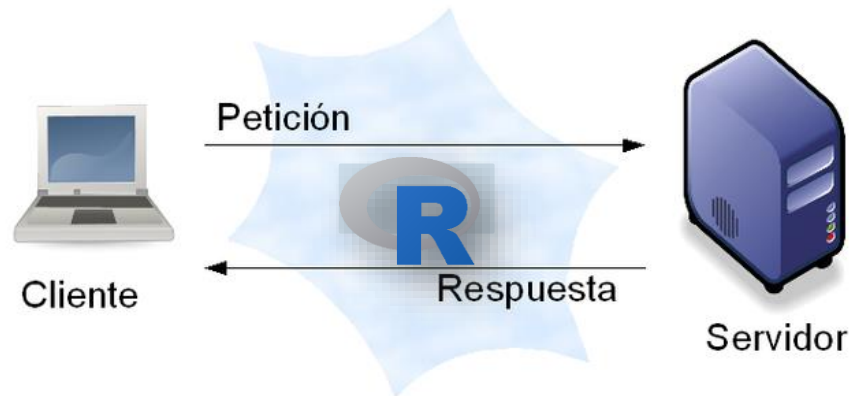
- Es gratis hasta 5 aplicaciones
- 25 horas activas



Server

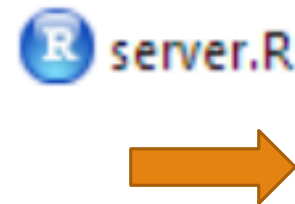
- Shiny Server Pro
- Shiny Server Open source
- Rstudio Conect

Arquitectura



Interfaz de usuario

- Controla la apariencia
- Widgets para usuario
- Representa resultado y gráficos



Servidor

- Instrucciones que, con los datos del usuario, producirá la salida deseada.

Estructura



The shortest viable shiny app

```
library(shiny)
ui <- fluidPage()

server <- function(input, output) {}

shinyApp(ui = ui, server = server)
```

Estructura



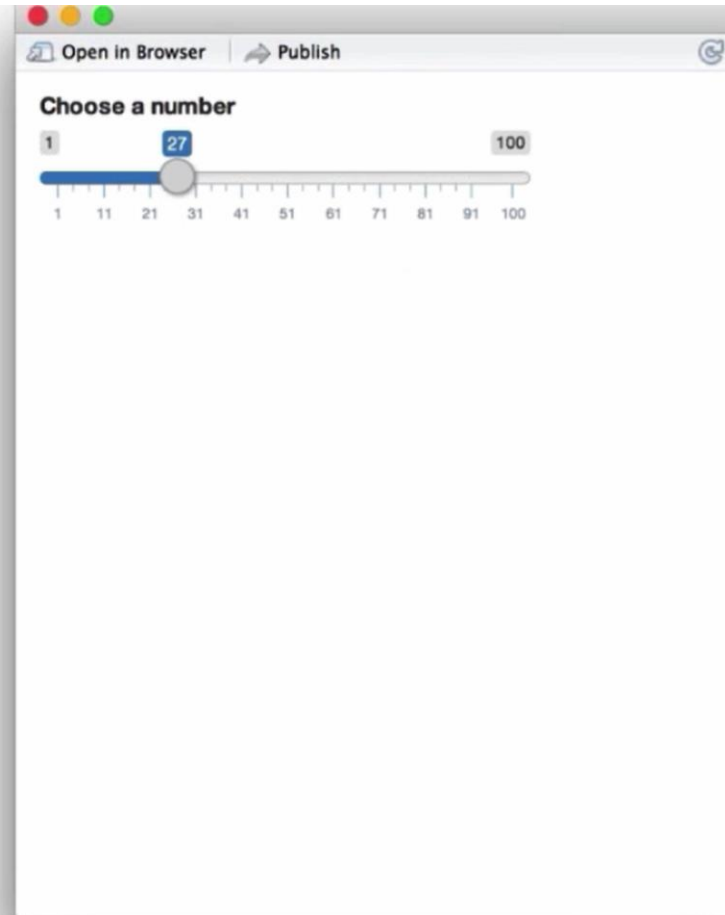
```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100)
)

server <- function(input, output) {

}

shinyApp(ui = ui, server = server)
```



Estructura



Button

Action

`actionButton()`

Single checkbox

☒ Choice A

`checkboxInput()`

Checkbox group

☒ Choice 1
☐ Choice 2
☐ Choice 3

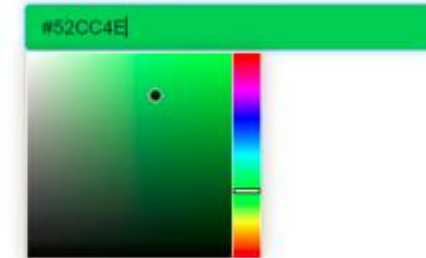
`checkboxGroupInput()`

Date input

2014-01-01

`dateInput()`

Colour input



`colourpicker::colourInput()`

Date range

2014-01-24 to 2014-01-24

`dateRangeInput()`

File input

Choose File No file chosen

`fileInput()`

Numeric input

1

`numericInput()`

Password Input

.....

`passwordInput()`

Radio buttons

☒ Choice 1
☐ Choice 2
☐ Choice 3

`radioButtons()`

Select box

Choice 1

`selectInput()`

Sliders



`sliderInput()`

Text input

Enter text...

`textInput()`

Text area

Multiple lines
of text

`textAreaInput()`

Estructura



```
library(shiny)

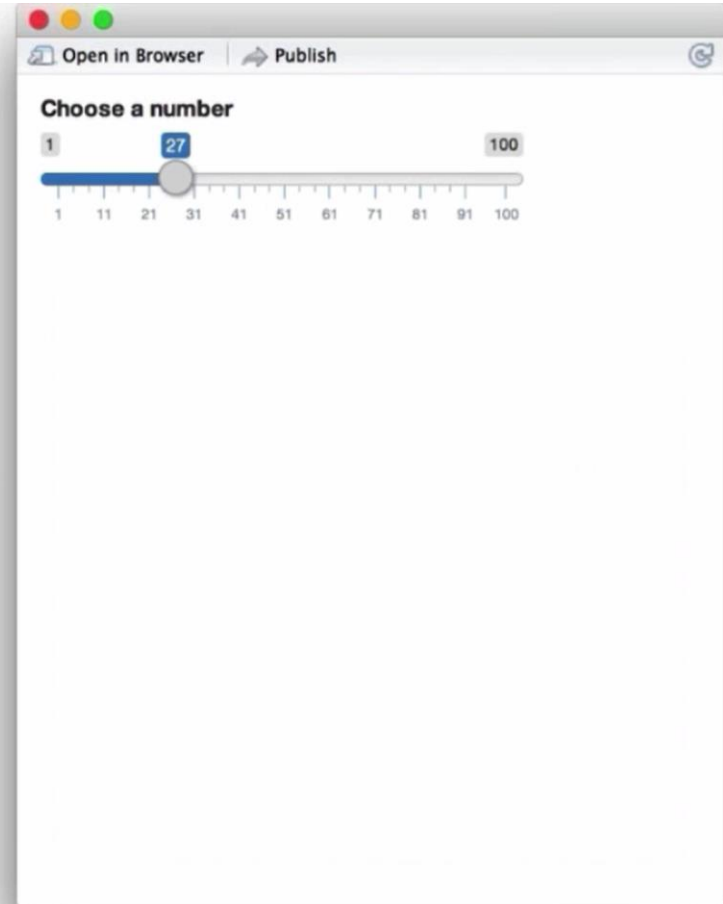
ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <-

}

shinyApp(ui = ui, server = server)
```

1



Estructura



```
library(shiny)

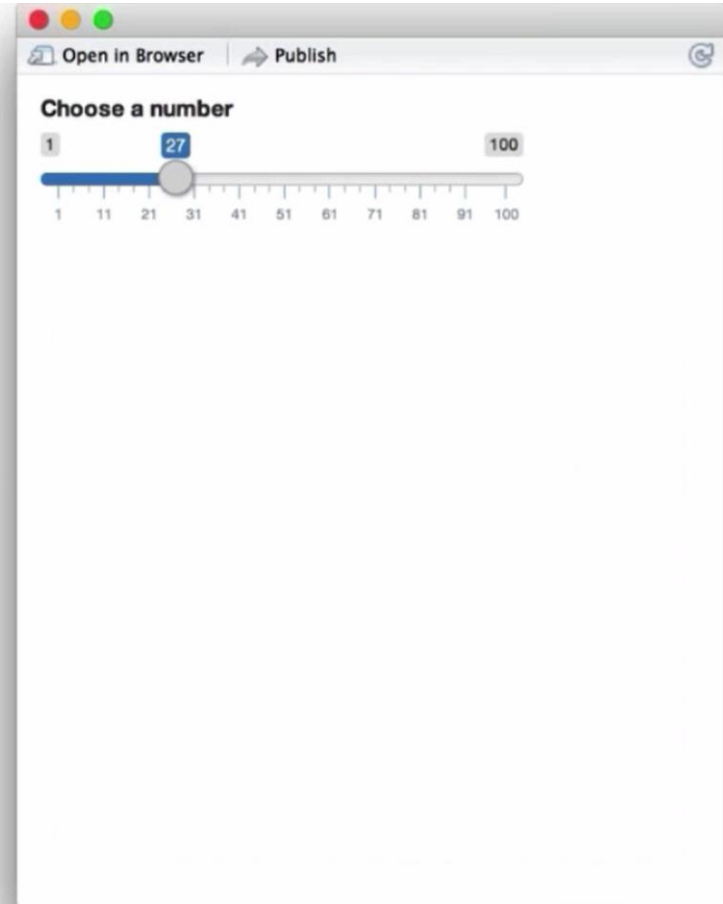
ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <-

}

shinyApp(ui = ui, server = server)
```

1



Estructura



```
library(shiny)

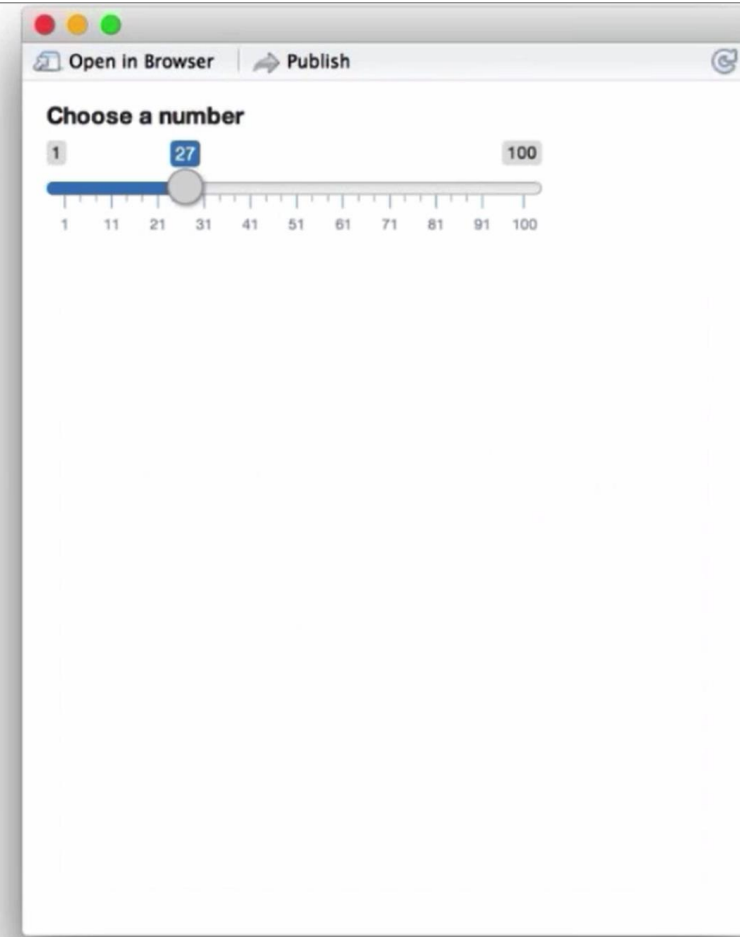
ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({

  })
}

shinyApp(ui = ui, server = server)
```

2



Estructura



function	creates
<code>renderDataTable()</code>	An interactive table <small>(from a data frame, matrix, or other table-like structure)</small>
<code>renderImage()</code>	An image (saved as a link to a source file)
<code>renderPlot()</code>	A plot
<code>renderPrint()</code>	A code block of printed output
<code>renderTable()</code>	A table <small>(from a data frame, matrix, or other table-like structure)</small>
<code>renderText()</code>	A character string
<code>renderUI()</code>	a Shiny UI element

Estructura



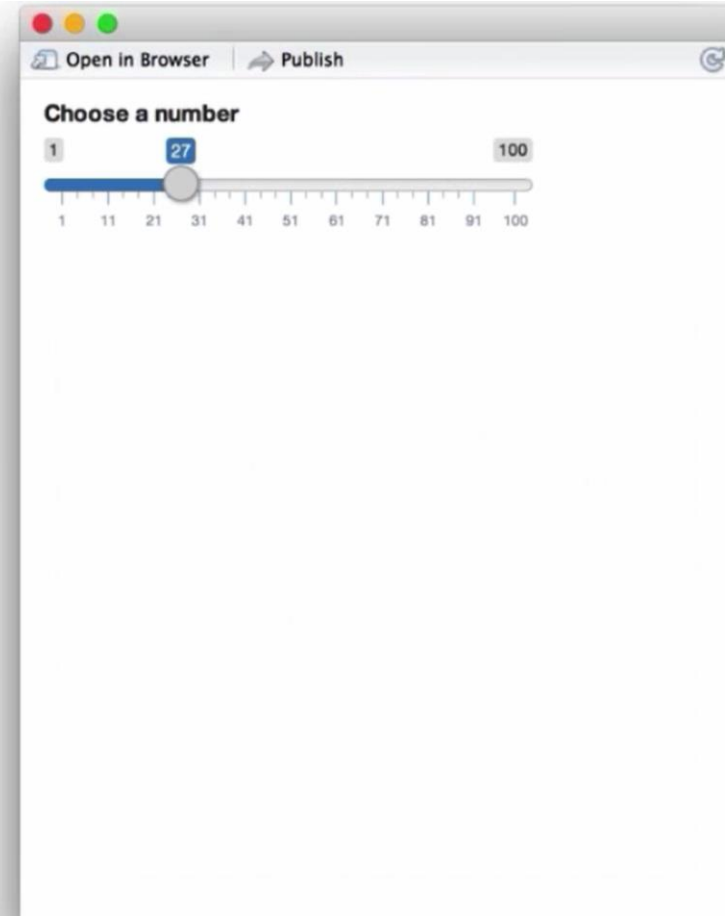
```
library(shiny)

ui <- fluidPage(
  sliderInput(inputId = "num",
    label = "Choose a number",
    value = 25, min = 1, max = 100),
  plotOutput("hist")
)

server <- function(input, output) {
  output$hist <- renderPlot({
    hist(rnorm(input$num))
  })
}

shinyApp(ui = ui, server = server)
```

3



Reactividad

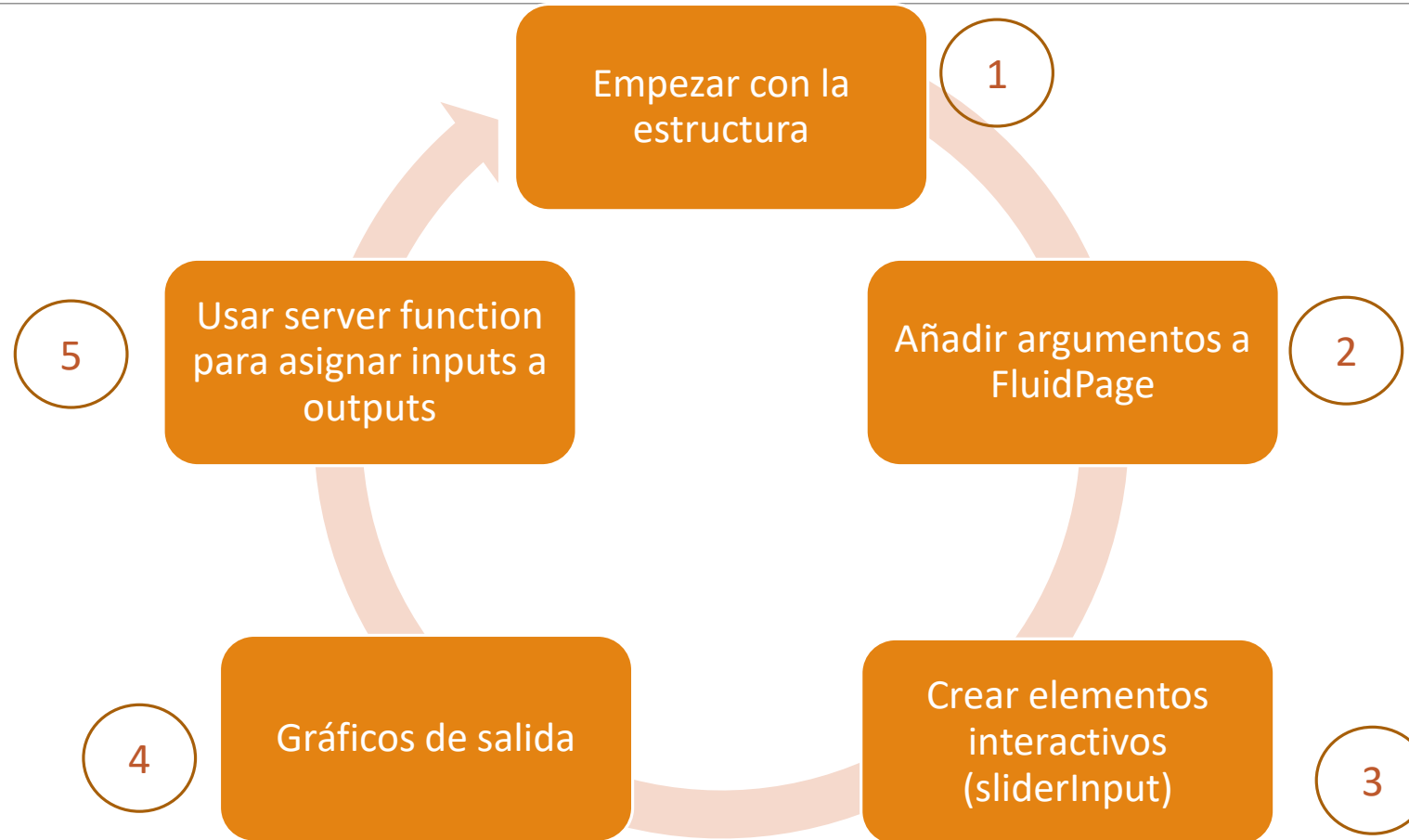


- Valores reactivos >>> input
- Solo puedes llamar a los valores reactivos desde una función diseñada para ella.
- Los valores reactivos notifican , y los objetos creados por las funciones reactivas responden a cualquier cambio de los valores reactivos.

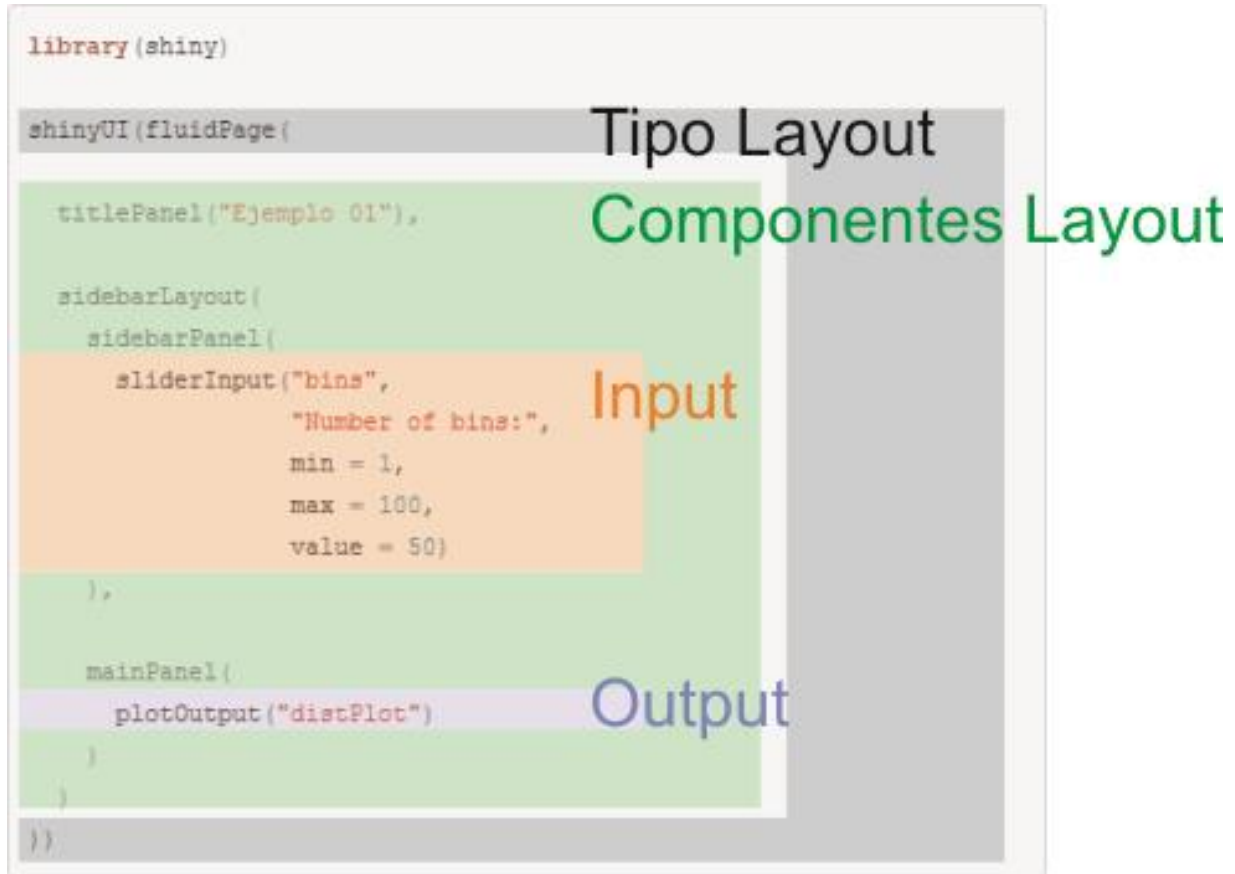
OBSERVE

ISOLATE

Pasos a seguir



Partes de la aplicación



Paquetes



Propios de Shiny:

- Shinydashboard
- Flexdashboard

<https://rmarkdown.rstudio.com/flexdashboard/examples.html>

Gráficos:

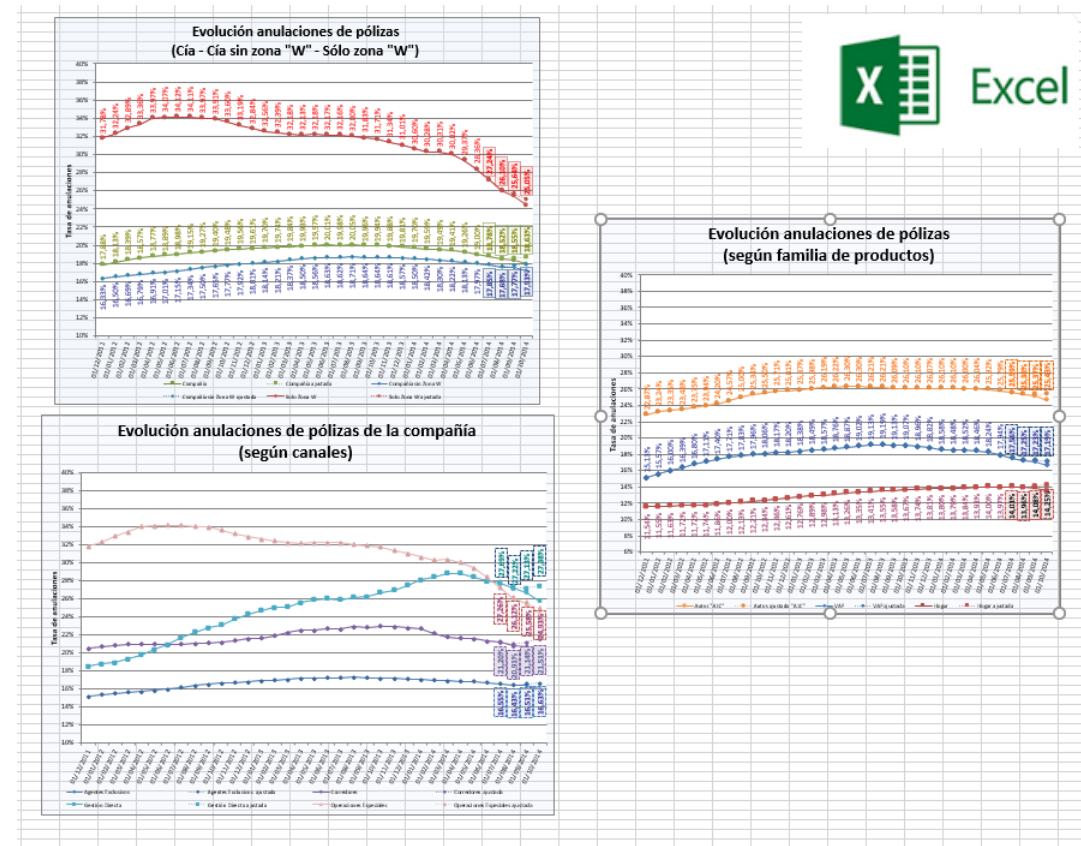
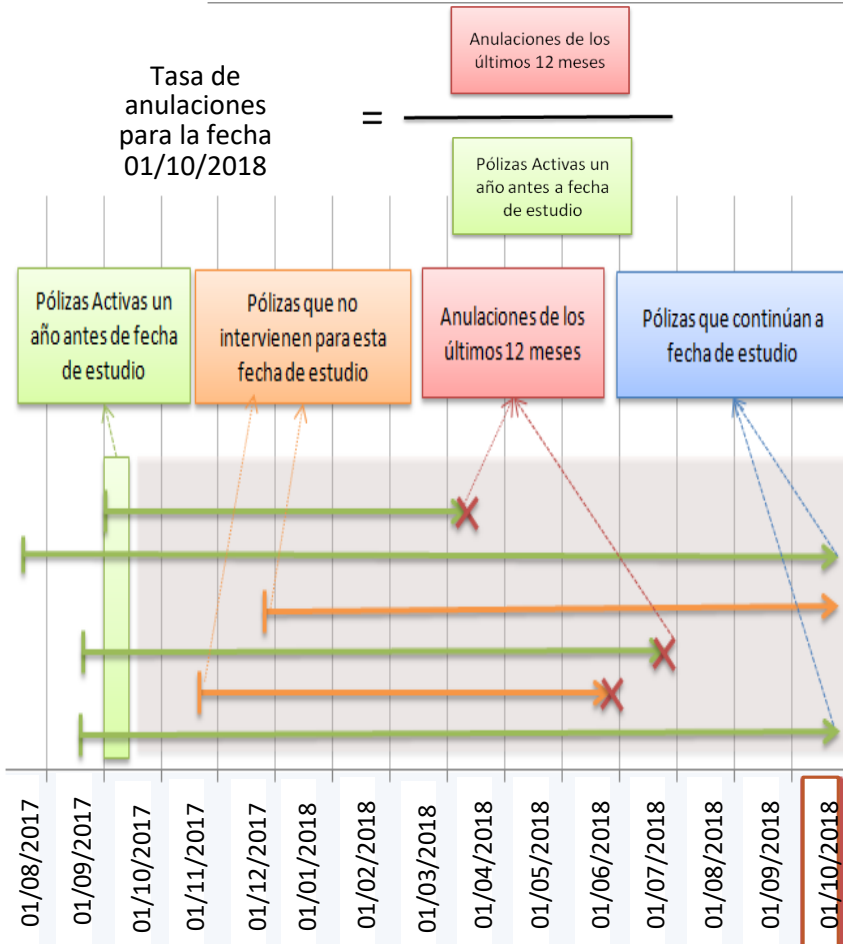
- Plotly
- Highcharter
- rCharts

Manipulación:

- Reshape2
- Data.table
- Dplyr
- lubridate



Ejemplo – Cuadro de mandos Anulaciones



Preguntas



Gracias!

- ijtudoramirez@hotmail.com
- ignacio-jose.tudo@helvetia.es

