

Material Imprimible

Curso de Análisis de datos con R

## Módulo VI

### **Contenidos:**

- ¿Qué es R Shiny?
- Estructura de un Shiny App
- Cómo visualizar un tablero de shiny
- Donde publicar un tablero shiny

## ¿Qué es R Shiny?

R Shiny es un marco de desarrollo web para el lenguaje de programación R. Shiny fue desarrollado por RStudio y permite a los usuarios crear aplicaciones web interactivas directamente desde R.

Con Shiny, los usuarios pueden convertir sus análisis de datos en aplicaciones web interactivas sin necesidad de conocimientos extensivos de programación web ni diseño. Esto facilita la comunicación de resultados y visualizaciones de datos a un público más amplio, ya que los usuarios pueden interactuar con los datos y ajustar parámetros de manera dinámica. De lo contrario, los objetos quedan dentro de R Studio y hay que exportarlos manualmente para uso, como los plots realizados, perdiendo así, la interactividad.

Estas aplicaciones permiten exponer resultados de procesos y gráficos, pero también armar interfaces interactivas para que diversos usuarios interactúen con el aplicativo.

Para empezar a usarlo, hay que instalar el paquete correspondiente:

```
install.packages("shiny")
```

## Estructura de un Shiny App

Las apps de Shiny se componen de dos partes principales: el UI y el server

UI (Interfaz de Usuario): La interfaz de usuario define cómo se ve la aplicación. Puedes crear elementos como botones, gráficos, cuadros de texto, etc., para que los usuarios interactúen

Server: El servidor maneja la lógica y las operaciones de cálculo en la aplicación. Responde a las interacciones del usuario y actualiza dinámicamente la interfaz de usuario en consecuencia.

El contenido de cada parte puede guardarse en un único script llamado App.R, o en distintos archivos llamados ui.R y server.R

Es importante ser ordenados y precisos en cómo diseñamos nuestro código, ya que si no separamos bien la interfaz de los algoritmos lógicos, nuestra aplicación puede fallar a medida que se le usen las funcionalidades.

El siguiente código es un ejemplo de la página oficial de Shiny App (<https://shiny.posit.co/r/getstarted/shiny-basics/lesson1/index.html>)

```
library(shiny)

# Define UI for app that draws a histogram ----
ui <- fluidPage(

  # App title ----
  titlePanel("Hello Shiny!"),

  # Sidebar layout with input and output definitions ----
  sidebarLayout(

    # Sidebar panel for inputs ----
    sidebarPanel(

      # Input: Slider for the number of bins ----
      sliderInput(inputId = "bins",
                  label = "Number of bins:",
                  min = 1,
                  max = 50,
                  value = 30)

    ),

    # Main panel for displaying outputs ----
    mainPanel(

      # Output: Histogram ----
```

```
    plotOutput(outputId = "distPlot")

  )
)

# Define server logic required to draw a histogram ----
server <- function(input, output) {

  # Histogram of the Old Faithful Geyser Data ----
  # with requested number of bins
  # This expression that generates a histogram is wrapped in a call
  # to renderPlot to indicate that:
  #
  # 1. It is "reactive" and therefore should be automatically
  #    re-executed when inputs (input$bins) change
  # 2. Its output type is a plot
  output$distPlot <- renderPlot({

    x <- faithful$waiting
    bins <- seq(min(x), max(x), length.out = input$bins + 1)

    hist(x, breaks = bins, col = "#007bc2", border = "white",
         xlab = "Waiting time to next eruption (in mins)",
         main = "Histogram of waiting times")

  })

}
```

Una vez que desarrollamos nuestro UI y nuestro server, nos queda un archivo App.R con esta estructura:

```
library(shiny)
ui <- fluidPage(
  titlePanel("Mi Aplicación Shiny"),
  sidebarLayout(
    sidebarPanel(
```

***En este lugar, se insertan los objetos que queremos en el panel lateral. En el curso, incorporamos botones y filtros.***

```
),
  mainPanel(
```

***En este lugar, se insertan los objetos que queremos en el panel principal. En el curso, incorporamos gráficos.***

```
)
)
)
```

```
server <- function(input, output) {
```

***En este lugar, se insertan las lógicas que necesitamos que se calculen en el app. En el curso, calculamos los datos que responden reactivamente a los filtros de usuario y abastecen el gráfico que se expone en el UI***

```
}
```

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

Una vez que tenemos este archivo predefinido, ejecutamos el app.R desde RunApp:

```
runApp("my_app") // Es importante que se introduzca toda la ruta del  
script /directorio/my_app
```

Una vez que la aplicación esté en ejecución, podrás interactuar con ella a través del navegador web. RStudio abrirá una pestaña nueva en tu navegador predeterminado con la dirección `http://127.0.0.1:port/` (tu entorno local)

## Donde publicar un tablero shiny

Existen varias opciones para publicar y alojar aplicaciones Shiny de forma gratuita. Estas suelen tener ciertas limitaciones en cuanto a recursos y capacidad de tráfico, por lo que si la aplicación Shiny se vuelve más grande o requiere mayores recursos, posiblemente haya que utilizar algún plan que requiera pago digital. Recomendamos siempre revisar la confiabilidad del sitio antes de contratar algún servicio.

También es importante revisar si la aplicación y los datos que se publican cumplen los Términos, condiciones y políticas del servicio.

**Shinyapps.io:** Es un servicio en la nube proporcionado por RStudio específicamente para alojar aplicaciones Shiny. Ofrece una capa gratuita con restricciones, pero puede ser suficiente para aplicaciones más pequeñas o para propósitos de prueba y demostración.

Sitio web: **[shinyapps.io](http://shinyapps.io)**