

Shiny de RStudio

Framework de R para aplicaciones Web

Briggette Olenka Román Huaytalla

Facultad de Ciencias

Grupo Estudiantil ACECOM

Universidad Nacional de Ingeniería

1 de Julio de 2016

Índice

- 1 Introducción
 - 2 ¿Cómo empezar?
 - 3 IDE
- RStudio
 - Trabajando con Shiny
 - Trabajando con Shiny -
División del trabajo

Índice

1 Introducción

2 ¿Cómo empezar?

3 IDE

- RStudio
- Trabajando con Shiny
- Trabajando con Shiny - División del trabajo

Introducción

Shiny es un framework de RStudio con el cual nos permite hacer aplicaciones web, pudiendo así trabajar de manera dinámica con los datos para un presentación o para que el usuario final le facilite la manera de ingresar o obtener resultados de la bases de datos procesadas.



Índice

1 Introducción

2 ¿Cómo empezar?

3 IDE

- RStudio
- Trabajando con Shiny
- Trabajando con Shiny - División del trabajo

¿Cómo empezar?

Primero es instalar el paquete *shiny*:

```
>> install.package("shiny")
```

Tener cuidado que repositorio se elija para descargar ya que algunos no funcionan correctamente en este caso se usó el *repositorio de España Madrid*.

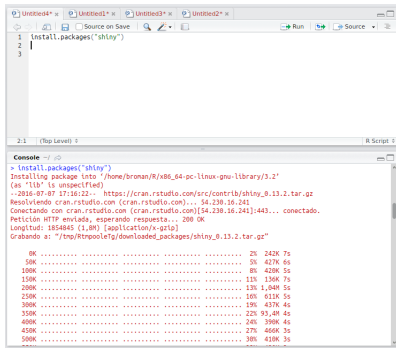
*** Esto es en caso de que se este trabajando desde terminal*

¿Cómo empezar? - Parte 2

Ahora trabajando desde el IDE de Rstudio primero es instalar el paquete *shiny*:

```
> install.package("shiny")
```

Que al ejecutar nos mostrara en la consola del editor como se muestra en la imagen.



The screenshot shows the RStudio interface with the R console open. The command `install.packages("shiny")` has been executed. The console output shows the progress of the installation, including the download of the package from CRAN and the progress of the installation.

```
install.packages("shiny")
Installing package into '/home/broman/R/x86_64-pc-linux-gnu-library/3.2'
(as 'lib' is unspecified)
--2016-07-07 17:16:22-- https://cran.rstudio.com/src/contrib/shiny_0.13.2.tar.gz
Resolving cran.rstudio.com (cran.rstudio.com)... 54.238.16.241
Conectando con cran.rstudio.com (cran.rstudio.com)[54.238.16.241]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1854845 (1.8M) [application/x-gzip]
Grabando a: "/tmp/RtmpoolEg/downloaded_packages/shiny_0.13.2.tar.gz"

0K ..... 2% 242K 7s
50K ..... 5% 427K 6s
100K ..... 8% 420K 5s
150K ..... 11% 136K 7s
200K ..... 13% 1,04M 5s
250K ..... 16% 611K 5s
300K ..... 19% 437K 4s
350K ..... 22% 93,4M 4s
400K ..... 24% 390K 4s
450K ..... 27% 466K 3s
500K ..... 30% 410K 3s
```

Índice

1 Introducción

2 ¿Cómo empezar?

3 IDE

- RStudio
- Trabajando con Shiny
- Trabajando con Shiny - División del trabajo

Índice

1 Introducción

2 ¿Cómo empezar?

3 IDE

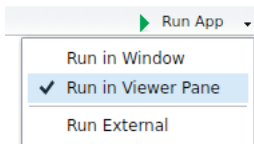
■ RStudio

■ Trabajando con Shiny

■ Trabajando con Shiny - División del trabajo

¿Como ejecutar?

La manera en como trabaja shiny nos permitirá ver ejecución de nuestro trabajo en distintas maneras. Teniendo el siguiente panel en nuestro IDE.



- **Run in Window**, se ejecutara dentro del IDE en la ventana de trabajo.
- **Run in Viewer Pane**, se abrirá un panel externo para visualizar nuestra ejecución.
- **Run in external**, se ejecutara nuestro programa en un navegador web. En este caso *Chrome*

*** Porque las diferencias, debido a que así uno puede visualizar como quedara para el usuario.*

Índice

1 Introducción

2 ¿Cómo empezar?

3 IDE

- RStudio

- **Trabajando con Shiny**

- Trabajando con Shiny - División del trabajo

Empezando:

En todos nuestros programas que realizaremos usando el framework de shiny comenzaremos llamando a la librería de *shiny*

```
> library(shiny)
```

Shiny

Habiendo encabezado el programa con la librería **shiny** se empieza a armar el código con dos secciones una que es para la edición frontal (La parte en que se muestra en ejecución la interfaz de Usuario o UI) que sería el **fluidPage** y la parte que trabaja internamente que sería el **server**

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

server <- function(input, output){
  ...
}

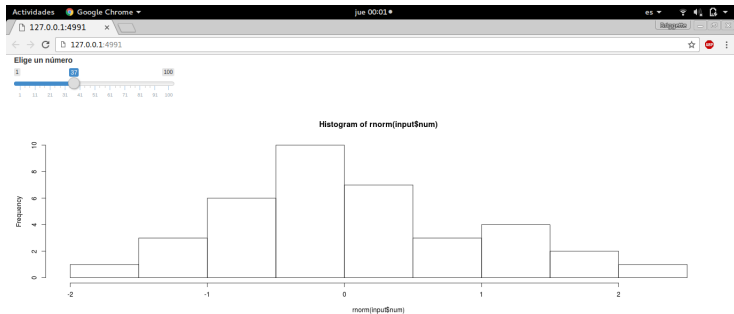
shinyApp(ui = ui, server = server)
```

Ejemplo

```
library(shiny)
ui <- fluidPage(
  sliderInput(inputId = "num",
    label = " Elige un número",
    value = 50, min = 1 , max=100),
  plotOutput("hist")
)
server <- function(input, output){
  output \textdollar hist<-renderPlot({
    title <-"Histograma de datos"
    hist(rnorm(input\textdollar num))
    main = title
  })
}
shinyApp(ui =ui,server = server)
```

Ejemplo: Ejecución

En el cual nos aparecerá algo como esto:



Índice

1 Introducción

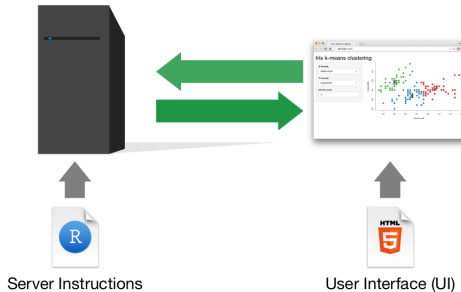
2 ¿Cómo empezar?

3 IDE

- RStudio
- Trabajando con Shiny
- Trabajando con Shiny - División del trabajo

¿Como trabaja shiny realmente?

Shiny trabaja dividiendo su trabajo en dos secciones en las cuales una es la parte de **UI** (*User interface*) y la otra es la de **server** (que es donde procesa la data obtenida o almacenada).



División del programa: Estructura

Ahora se trabajara con archivos independientes donde uno trabajara la parte de *FrontEnd* osea la **UI** y la parte de *BackEnd* que seria la parte de **Server**. Teniendo la estructura siguiente

Primer Archivo: *ui.R*

```
#ui.R
library(shiny)
shinyUI( fluidPage( ...
))
```

Segundo archivo: *server.R*

```
#server.R
library(shiny)
shinyServer(
  function(input, output){...
})
```

Ejemplo: ui.R

```
library(shiny)
shinyUI(
  fluidPage(
    titlePanel("Shiny App"),
    sidebarLayout(
      sidebarPanel(
        sliderInput(inputId = "num",
          label = "Elige un número",
          value = 50, min = 1, max = 100)
      ),
      mainPanel(
        plotOutput("distPlot")
      )
    )
  )
)
```

Ejemplo: `server.R`

```
library(shiny)
shinyServer(
  function(input, output){
    data <- reactive({
      rnorm(input$num)
    })
    output$distPlot <- renderPlot({
      hist(rnorm(input$num))
    })
  }
)
```

Formas de ejecución:

Ahora si bien puedes ejecutar tu programa en ese momento cuando estas editandolo, que pasa ahora que desees ejecutarlo externamente o llamarlo desde otro programa. Para ello debes haber guardado tus archivos *ui.R* y *server.R* en una carpeta o directorio, que en este caso estan en *Escritorio/primerAppShiny* , para poder correrlo de la siguiente manera:

```
> library(shiny)
> runApp('ubicacion/del/directorio/de/tu/app')
```

Ejemplo:

```
> library(shiny)
> runApp('Escritorio/primerAppShiny')
```