

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE  
HUAMANGA**

**FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL**

**ESCUELA DE FORMACIÓN PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**ASIGNATURA:**

**ESTADÍSTICA II**

**TEMA:**

**APLICACIONES WEB EN R CON SHINY: Acciones del Mercado de Valores**

**DOCENTE:**

**ROMERO PLASENCIA, Jackson Mcoy**

**ESTUDIANTES:**

**CÁRDENAS QUISPE, Arístides (27130649)**

**JORGE MISARAYME, Raquel Rebeca (27162118)**

**TRISTÁN QUISPE, Guadalupe (27162506)**

**AYACUCHO 2019**

## INTRODUCCIÓN

Existen muchas opciones para construir Aplicaciones Web y diversas arquitecturas Cliente-Servidor que facilitan este proceso, sin embargo, nos interesa transitar el camino para construir Aplicaciones Web desde R utilizando el paquete Shiny.

Normalmente es necesario contar con un programador que conozca sobre las tecnologías Web como HTML5/JavaScript/CSS3, sin embargo, es posible construir Aplicaciones Web dinámicas e interactivas mediante las facilidades que ofrece Shiny sin requerir mucho conocimiento sobre las tecnologías Web.

RStudio ha hecho grandes esfuerzos por facilitar el acceso y usabilidad de R como lenguaje y en este sentido ha promovido la creación de nuevos paquetes para manejar documentación y también crear contenido Web. En esa misma línea también crearon el paquete shiny. Las funciones de este paquete se ocupan de la comunicación entre el Cliente y el Servidor utilizando la programación reactiva (Reactive), además genera el HTML5/JavaScript/CSS necesario para construir las Aplicaciones Web

## CONCEPTOS

**R:** R es un entorno y lenguaje de programación con un enfoque al análisis estadístico.

R nació como una reimplementación de software libre del lenguaje S, adicionado con soporte para alcance estático. Se trata de uno de los lenguajes de programación más utilizados en investigación científica, siendo además muy popular en los campos de aprendizaje automático (machine learning), minería de datos, investigación biomédica, bioinformática y matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo y graficación.

**RStudio:** Es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo.

**Shiny:** es una herramienta para crear fácilmente aplicaciones web interactivas (apps) que permiten a los usuarios interactuar con sus datos sin tener que manipular el código.

La programación Reactiva enfatiza el uso de:

- Valores que cambian en el tiempo
- Expresiones que registran esos cambios
- Ver el ejemplo que hemos publicado en este mismo blog.

## APLICACIONES WEB EN R CON SHINY: Acciones del Mercado de Valores Americano

Una aplicación Shiny está conformada por un archivo `app.R` o dos archivos `ui.R` y `server.R`. Es decir, se puede hacer una aplicación Shiny partiendo de un solo archivo con todo el código o se puede partir de dos archivos que separan los aspectos de la interfaz de los aspectos centralizados (servidor).

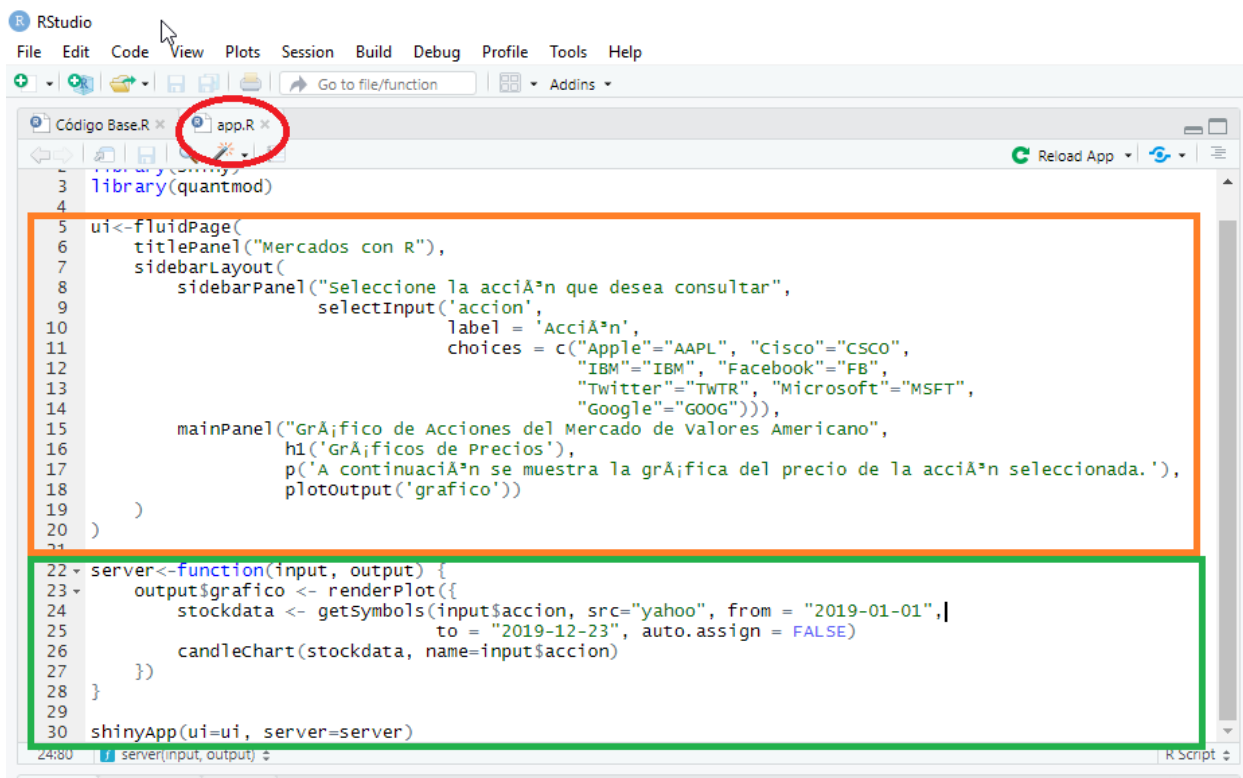
### Estructura:

- **app.R** es el archivo que contiene tanto los elementos de la interfaz como del servidor.
- **ui.R** es el archivo donde se especifican los elementos de la interfaz y la disposición de dichos elementos en la pantalla.

- **server.R** es el archivo donde se programa la lógica del servidor y se genera el contenido dinámico que depende de las interacciones con la pantalla.

Pero con la última actualización una aplicación web sólo se desarrolla en una sola interfaz.

Los elementos **ui.R** y **server.R** se encuentra en una misma interfaz.



## Por parte del usuario

Pondremos un título mediante la función **titlePanel()**, la creación de una pantalla con una barra lateral y un panel principal, mediante las funciones **sidebarPanel()** y **mainPanel()** respectivamente. Ambos componentes se colocan dentro de la llamada a la función **sidebarLayout()** que genera una página dividida en una barra lateral y un panel principal.

```

ui<-fluidPage(
  titlePanel("Mercados con R"),
  sidebarLayout(

```

En **sidebarPanel()** se agregan los elementos que van en el Panel separados por coma como sigue:

```
sidebarPanel("Seleccione la acción que desea consultar",
```

Para colocar un texto se agrega el texto entre comillas y luego el **selectInput()** con las opciones para seleccionar algunas de las acciones como sigue.

```
  selectInput('accion',  
    label = 'Acción',  
    choices = c("Apple"="AAPL", "Cisco"="CSCO",  
               "IBM"="IBM", "Facebook"="FB",  
               "Twitter"="TWTR", "Microsoft"="MSFT",  
               "Google"="GOOG"))),
```

Luego en el Panel principal vamos a agregar un texto descriptivo, un título con **h1()**, texto en un párrafo con **p()** y el gráfico con **plotOutput()**.

```
mainPanel("Gráfico de Acciones del Mercado de Valores Americano",  
  h1('Gráficos de Precios'),  
  p('A continuación se muestra la gráfica del precio de la acción seleccionada.'),  
  plotOutput('grafico'))
```

Entonces el código quedaría de esta forma por parte del usuario.

```
library(shiny)  
library(quantmod)  
  
ui<-fluidPage(  
  titlePanel("Mercados con R"),  
  sidebarLayout(  
    sidebarPanel("Seleccione la acción que desea consultar",  
      selectInput('accion',  
        label = 'Acción',  
        choices = c("Apple"="AAPL", "Cisco"="CSCO",  
                   "IBM"="IBM", "Facebook"="FB",
```

```

        "Twitter"="TWTR", "Microsoft"="MSFT",
        "Google"="GOOG"))),
    mainPanel("Gráfico de Acciones del Mercado de Valores Americano",
      h1('Gráficos de Precios'),
      p('A continuación se muestra la gráfica del precio de la acción seleccionada.'),
      plotOutput('grafico'))
  )
)

```

#### Nota:

1. Recordar que la librería **Shiny** viene con la aplicación web o puedes instalar antes de empezar el proyecto.
2. **Quantmod** es otro paquete o librería de R diseñado para asistir a los operadores cuantitativos (quantitative traders) en el desarrollo, pruebas y despliegue de modelos de trading basados en las estadísticas (también sirve para extraer datos).

#### Por parte del Servidor

En la interfaz hemos creado una variable input en la llamada a **selectInput()** denominada accion. En esta variable está almacenada la acción seleccionada. La forma de acceder a este valor desde el servidor es mediante la expresión **input\$accion**. Adicionalmente hemos creado una variable de salida en la llamada a **plotOutput()** denominada grafico, este es un gráfico que se genera en el servidor mediante la expresión **output\$grafico**. Es decir, que en el servidor utilizamos los **input\$** para generar los output\$.

A la variable **output\$grafico** debemos asignarle un gráfico en R. Esto se realiza mediante la función **renderPlot()**. Hay varias funciones de tipo renderXX para generar diversos tipos de salida. En este caso sólo queremos generar un gráfico.

Dentro de la llamada a **renderPlot()** el primer argumento es una expresión, es decir, una secuencia de comandos de R que producen el resultado esperado, en este caso un gráfico.

Para generar el gráfico vamos a usar las funciones del paquete **quantmod**. Lo primero que debemos hacer es incluir este paquete en nuestro programa mediante **library(quantmod)**. El código base es el siguiente:

```

server<-function(input, output) {
  output$grafico <- renderPlot({
    stockdata <- getSymbols(input$accion, src="yahoo", from = "2019-01-01",
                           to = "2019-12-23", auto.assign = FALSE)
    candleChart(stockdata, name=input$accion)
  })
}

```

Ahora teniendo las dos estructuras juntas tendremos el código en general:

```

library(shiny)
library(quantmod)

ui<-fluidPage(
  titlePanel("Mercados con R"),
  sidebarLayout(
    sidebarPanel("Seleccione la acción que desea consultar",
      selectInput('accion',
        label = 'Acción',
        choices = c("Apple"="AAPL", "Cisco"="CSCO",
                    "IBM"="IBM", "Facebook"="FB",
                    "Twitter"="TWTR", "Microsoft"="MSFT",
                    "Google"="GOOG"))),
    mainPanel("Gráfico de Acciones del Mercado de Valores Americano",
      h1('Gráficos de Precios'),
      p('A continuación se muestra la gráfica del precio de la acción seleccionada.'),
      plotOutput('grafico'))
  )
)

server<-function(input, output) {
  output$grafico <- renderPlot({
    stockdata <- getSymbols(input$accion, src="yahoo", from = "2019-01-01",
                           to = "2019-12-23", auto.assign = FALSE)
    candleChart(stockdata, name=input$accion)
  })
}

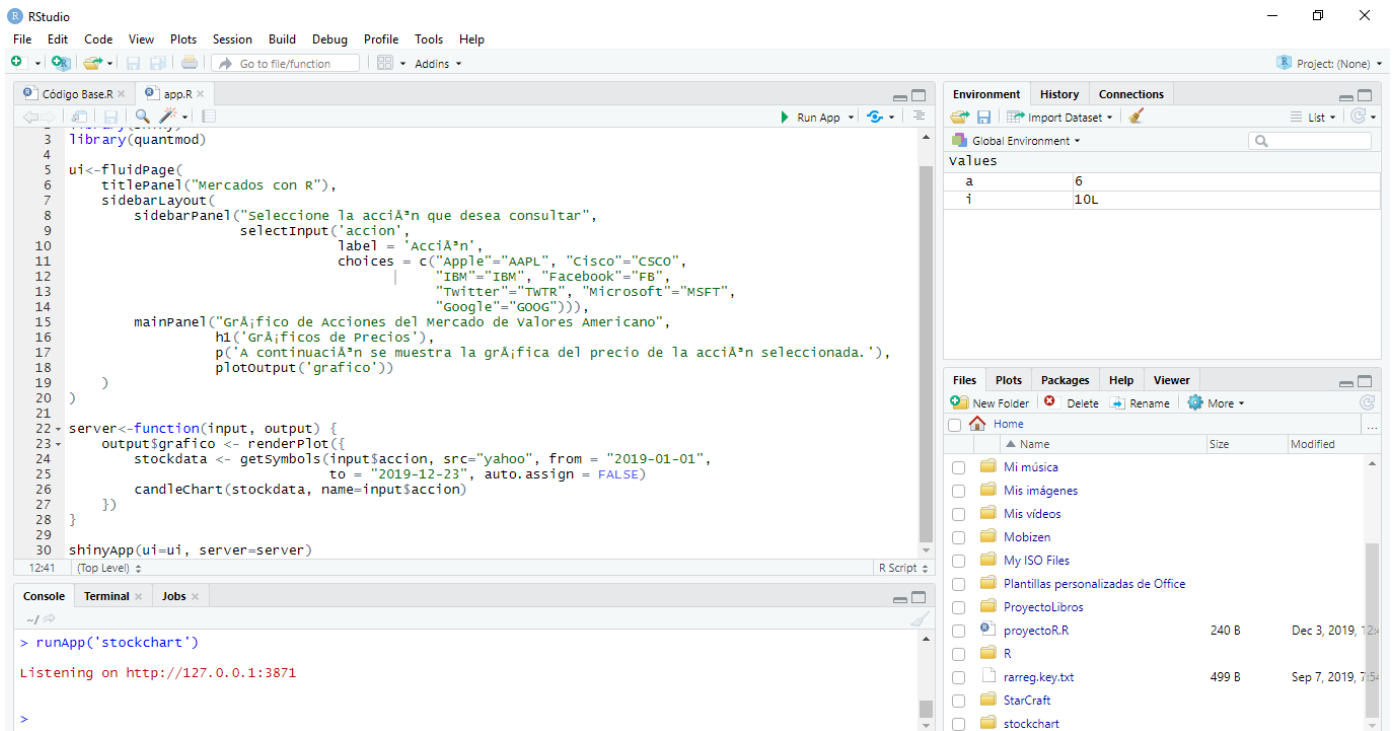
shinyApp(ui=ui, server=server)

```

**NOTA:** shinyApp(ui=ui, server=server) Es para la ejecución de la aplicación de ui.R y server.R juntamente.

## Capturas de la aplicación

### Código de ejecución



The screenshot shows the RStudio interface with the following components:

- Editor:** Contains the R code for the Shiny application. The code defines a UI with a sidebar for selecting a stock and a main panel for displaying a candlestick chart. The server logic fetches stock data from Yahoo and renders the chart.
- Console:** Shows the command `runApp('stockchart')` and the output `Listening on http://127.0.0.1:3871`.
- Environment:** Displays the global environment with variables `a` (value 6) and `i` (value 10L).
- Files:** Shows the project file structure, including folders like `Mi música`, `Mis imágenes`, and `stockchart`.

### Ejecución de los gráficos



The screenshot shows the Shiny application running in a web browser at `http://127.0.0.1:3871`. The application has the following layout:

- Title:** Mercados con R
- Input:** A dropdown menu labeled "Seleccione la acción que desea consultar" with "Apple" selected.
- Text:** Gráfico de Acciones del Mercado de Valores Americano
- Section Header:** Gráficos de Precios
- Text:** A continuación se muestra la gráfica del precio de la acción seleccionada.
- Figure:** A candlestick chart for AAPL from 2019-01-02 to 2019-12-20. The chart shows the price movement with a volume bar at the bottom. The last price is 279.440002 and the volume is 68,994,500.



## **Webgrafía**

<https://www.r-bloggers.com/lang/spanish/2818>

[https://es.wikipedia.org/wiki/R\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/R_(lenguaje_de_programaci%C3%B3n))

<https://synergy.vision/>

<https://cran.r-project.org/web/packages/quantmod/index.html>

<https://community.rstudio.com/t/error-install-packages-failed-to-lock-directory/40018>

[https://r-forge.r-project.org/R/?group\\_id=125](https://r-forge.r-project.org/R/?group_id=125)

<https://stackoverflow.com/questions/49360403/installation-of-quantmod-in-r>

<https://github.com/joshuaulrich/quantmod>

<https://rstudio.com/resources/cheatsheets/>