



# Introducción a R - Shiny

LNPP - Tech Noviembre, 2019

#### M.C. JORGE JUVENAL CAMPOS FERREIRA.

Asistente de investigación. Laboratorio Nacional de Políticas Públicas CIDE

#### Introducción



Shiny es un paquete de R que permite a los usuarios de este lenguaje de programación la creación de aplicaciones web interactivas.

Permite tomar tu trabajo creado en R y exponerlo mediante un navegador web de tal manera en que cualquier persona pueda revisarlo y visualizarlo sin tener necesidad de programar.

### Ventajas y pros



- Permite la creación de aplicaciones web sin tener un conocimiento profundo de HTML, CSS y JavaScript (los lenguajes de la web).
- Facilita el flujo de interactividad entre elementos, permitiendo la creación de elementos reactivos a las acciones del usuario.
- Permite el diseño de interfaces de usuario mediante funciones que traducen código de R a HTML, CSS y JavaScript.

#### Usos



- Crear dashboards (páginas web que llevan el registro de algo).
- Reemplazar cientos de páginas de pdf, permitiendo que el usuario seleccione las gráficas y los datos que exactamente requiere.
- Para dar a conocer resultados de investigaciones y bases de datos interesantes.
- Ayuda en la exploración rápida de bases de datos por parte de equipos grandes.
- Aplicaciones web en producción; dashboard para empresas, etc.

### Usos



### En palabras de Hadley Wickham:

"Shiny gives you the ability to pass on some of your R superpowers to anyone who can use a web browser"



### Alternativas sin R

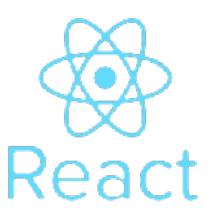


- Alternativas para usuarios de Python: Flask, Dash, Django.
- Alternativas para no programadores: Tableau.
- Alternativas para programadores tradicionales: Programación web con HTML, CSS, JavaScript (y sus frameworks React o Vue.js) en el Front y PHP o Django para el Back.









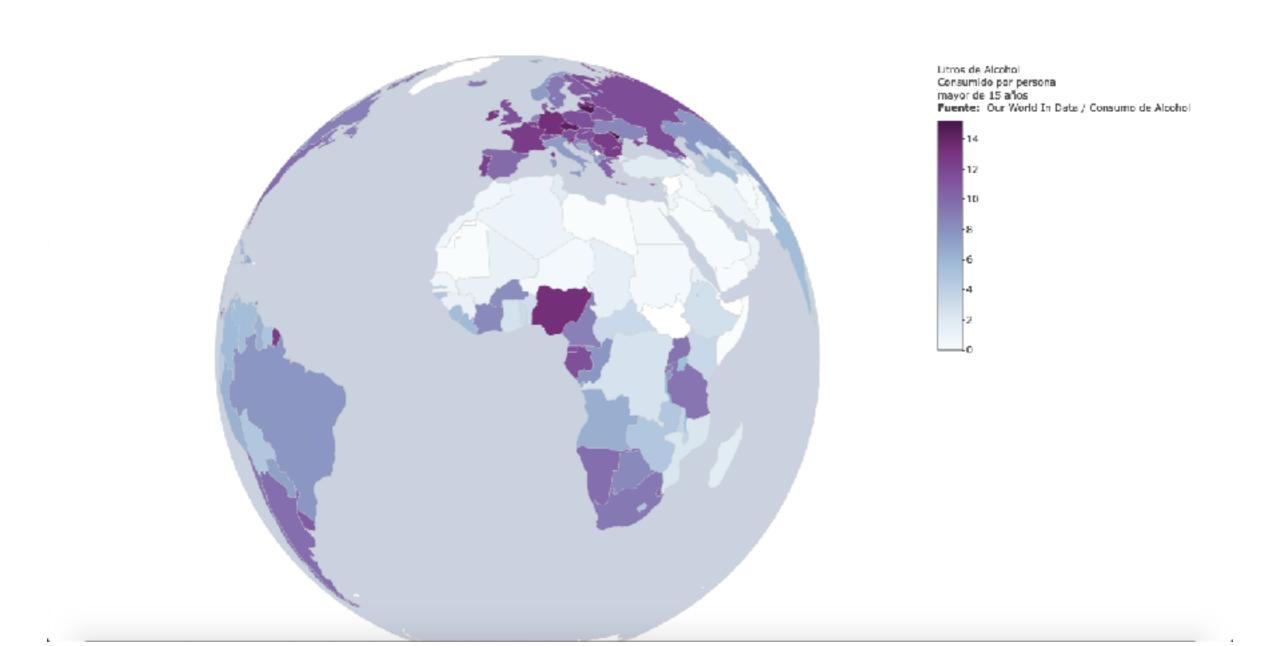




https://Inppmicrositio.shinyapps.io/metricadegobiernoabierto2019/

Dia 20 del #30DayMapChallenge Síganme en TW: @juvenalcamposf

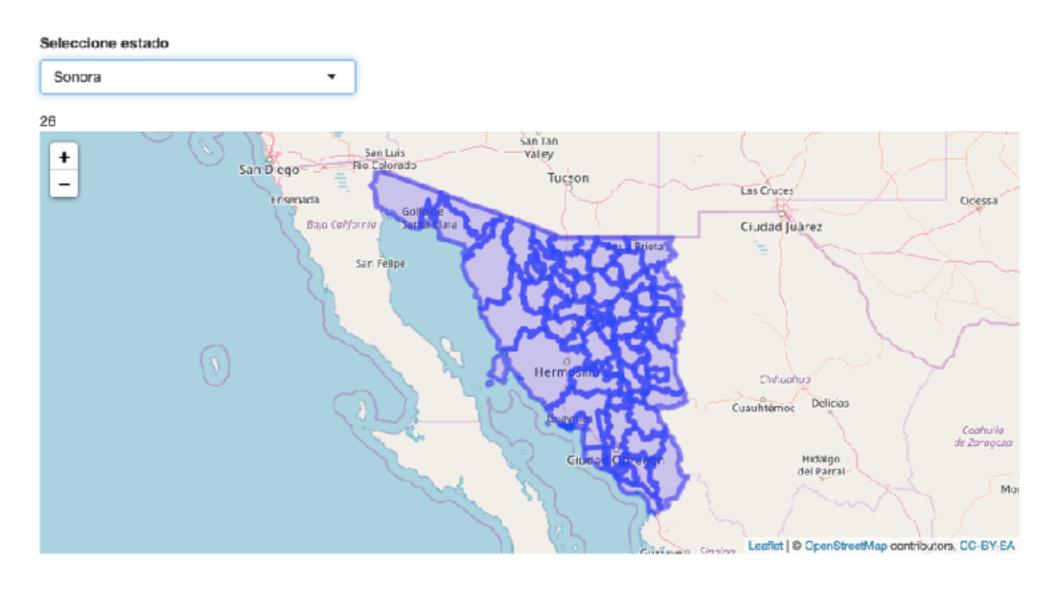




https://juvecampos.shinyapps.io/shinyAlcohol/



#### **Tutorial 1.**



http://rpubs.com/Juve\_Campos/Shiny\_Tutorial\_1



Consula disponibilidad. Biddelas — Modelo Precictivo de Disponibilidad de Bidide as



#### Ecobici

Ecobici es un sistema de bicicletas compartidas lanzado en febrero de 2010 por el gobierno de la Ciudad de México para la realización de viajes de corta distancia y duración a precios reducidos o de forma gratuita. El sistema quenta con 480,

#### http://appecobici.jjsantoso.com



## Manos a la obra

Paso 0: Descargar la librería shiny



Para instalar shiny utilizamos el código:

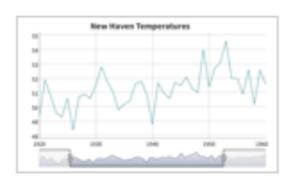
install.packages ("shiny")
#(Igual que con todas las demás librerías)

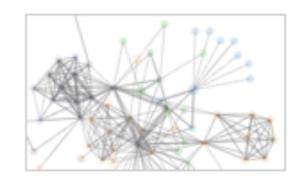


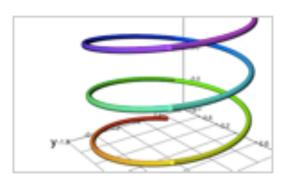
### Otras librerías que se llevan bien con shiny:

- htmlwidgets, para hacer visualizaciones dinámicas,
- Highcharter, para visualizaciones dinámicas,
- DT, para visualizar tablas con filtros y funcionalidades,
- leaflet, para hacer mapas interactivos y fáciles,
- ggplot2, para visualización de datos como imágenes
- plotly, para visualización de gráficas con tooltips
- shinydashboard, para estructura en forma de dashboard



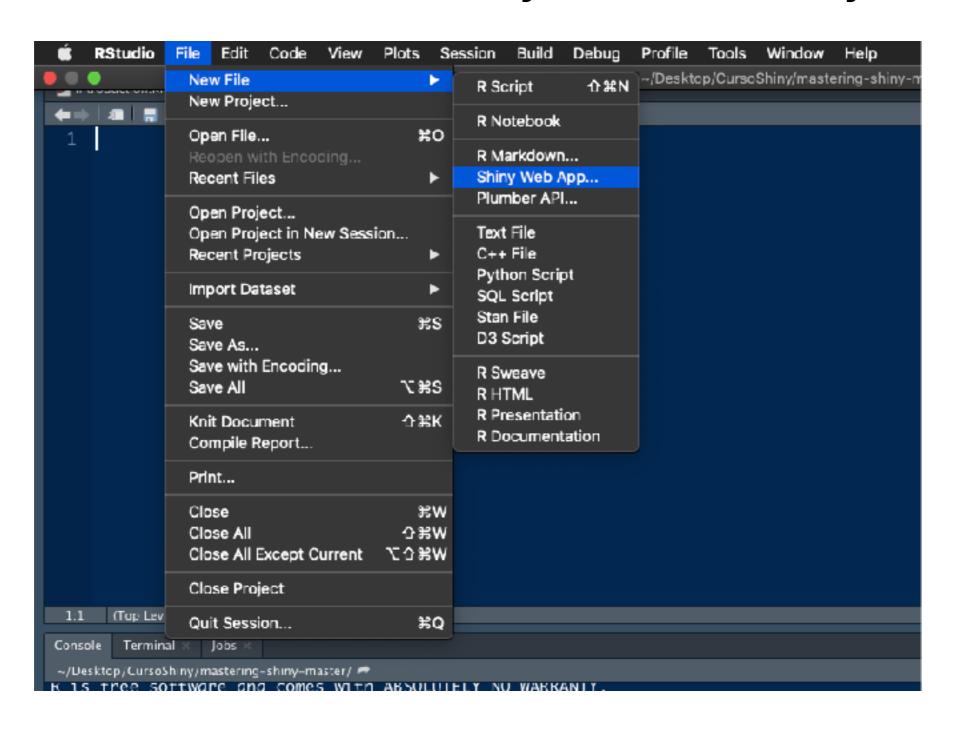






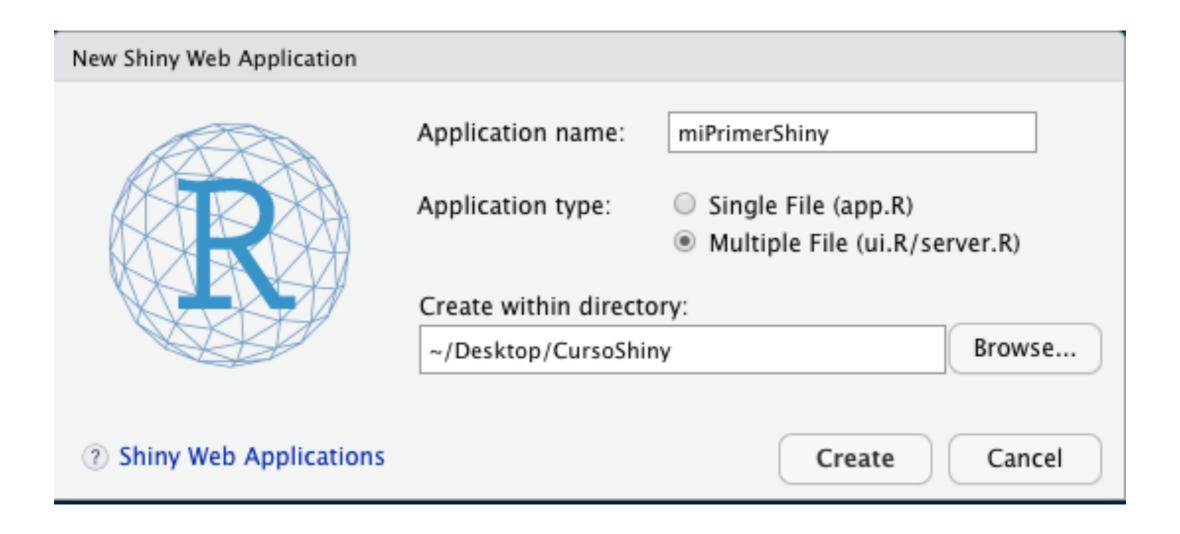


#### Paso 1: Ir a la ventana de archivo y seleccionar Shiny Web App



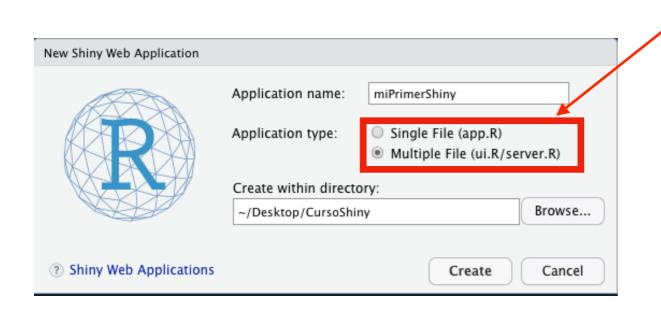


#### Paso 2: Seleccionar el nombre del proyecto y la ubicación





#### Paso 2: Seleccionar el nombre del proyecto y la ubicación



#### Single File:

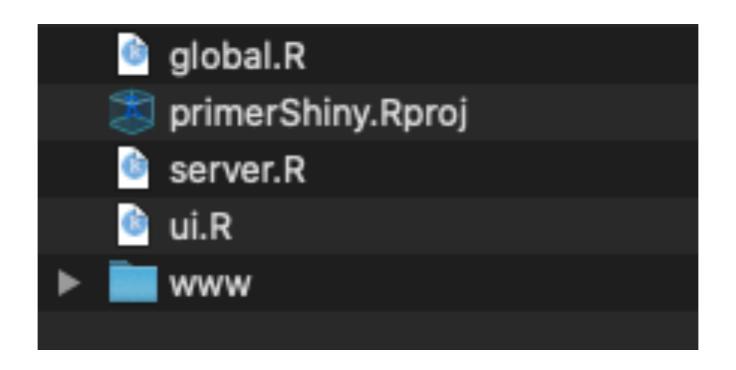
Recomendado para aplicaciones pequeñas (< 200 lineas de código).

#### **Multiple File:**

Recomendado para aplicaciones grandes.



Paso 2: Crear el archivo global. R para guardar variables globales, y una carpeta llamada www para guardar cosas (bases de datos, imágenes, código HTML, CSS, etc). Igualmente, cerciorarse de que exista un archivo . Rproj para el control de las rutas de archivos.

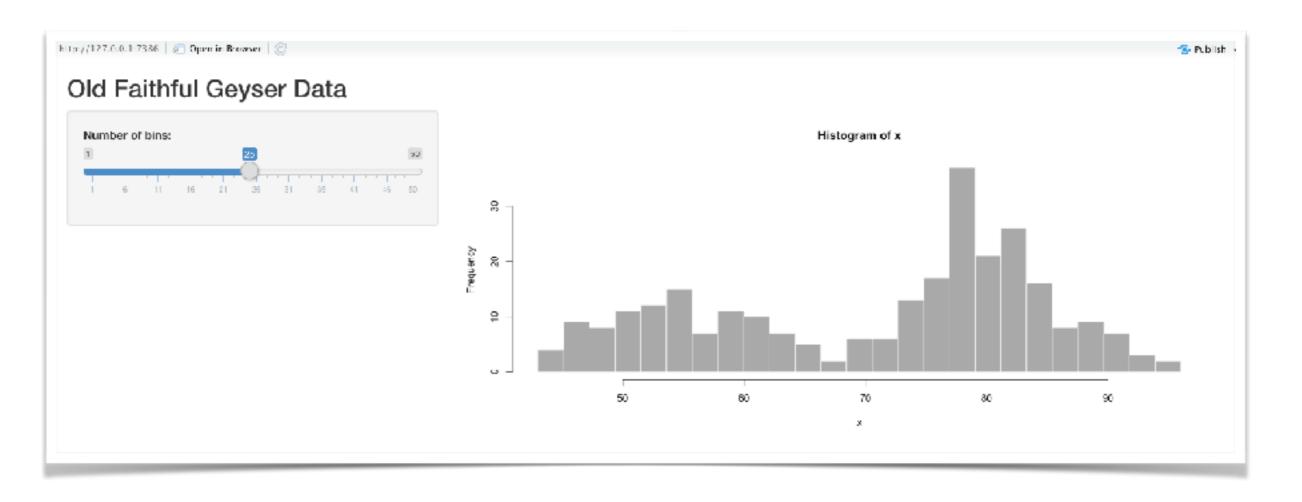


Directorio de archivos dentro de un shiny.



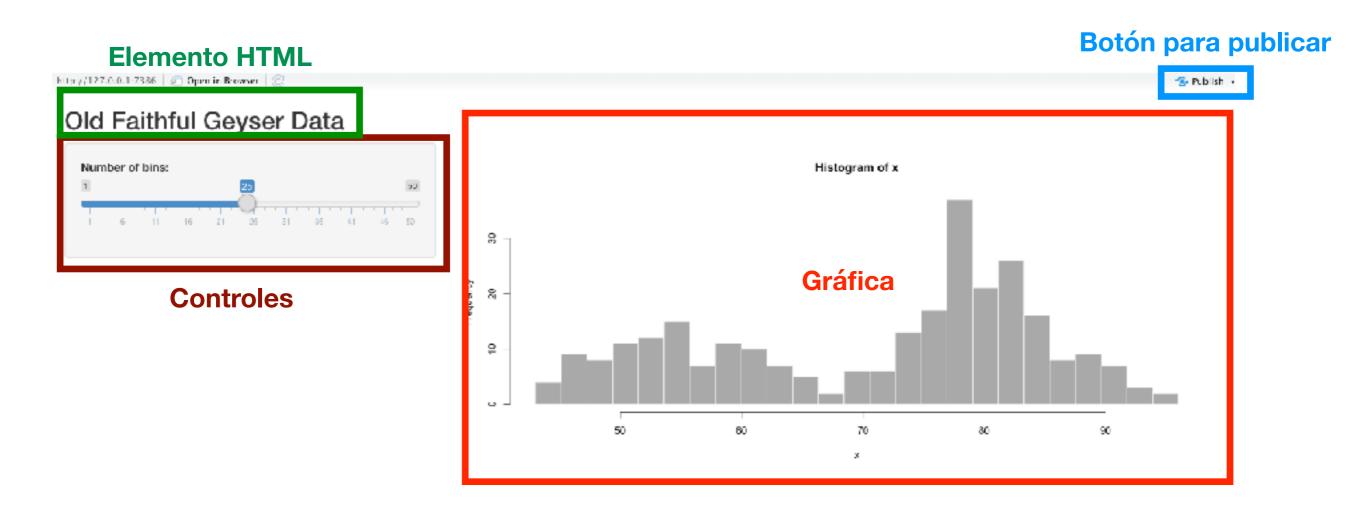


Cuando creamos un shiny, RStudio nos genera un shiny de prueba ejecutable, el cual se ve de la siguiente manera:





#### Analizando, vemos los siguientes elementos:





Ui o interfaz de usuario: Es la parte de la shinyapp con la que interactúa el usuario.

```
ui <- fluidPage(
    # Application title
    titlePanel("Old Faithful Geyser Data"),
    # Sidebar with a slider input for number of bins
    sidebarLayout(
        sidebarPanel(
            sliderInput("bins",
                        "Number of bins:",
                        min = 1,
                        max = 50,
                        value = 30)
        ),
        # Show a plot of the generated distribution
        mainPanel(
           plotOutput("distPlot")
```



Ui o interfaz de usuario: Es la parte de la shinyapp con la que interactúa el usuario.

```
ui <- fluidPage(
    # Application title
    titlePanel("Old Faithful Geyser Data"),
    # Sidebar with a slider input for number of bins
    sidebarLayout(
        sidebarPanel(
            sliderInput("bins",
                        "Number of bins:",
                        min = 1,
                        max = 50,
                        value = 30)
        ),
        # Show a plot of the generated distribution
        mainPanel(
           plotOutput("distPlot")
```

Controles, le mandan la instrucción al server para que este prepare la gráfica que el usuario quiere.



**server:** es el conjunto de funciones que crean los objetos que se van a visualizar en la interfaz de usuario. Esta es una función que tienen anidadas funciones más pequeñas, que son las que crean el código.

```
# Define server logic required to draw a histogram
server <- function(input, output) {</pre>
    output$distPlot <- renderPlot({</pre>
        # generate bins based on input$bins from ui.R
             <- faithful[, 2]
        bins <- seq(min(x), max(x), length.out = input$bins + 1)
        # draw the histogram with the specified number of bins
        hist(x, breaks = bins, col = 'darkgray', border = 'white')
    })
```



Los argumentos son input y output. **Input** le da a las funciones del server los ingredientes para poder funcionar, y **output** le entrega al ui los objetos ya preparados.

```
# Define server logic required to draw a histogram
server <- function(input, output) {</pre>
        Preparado
    output$distPlot <- renderPlot({
        # generate bins based on input$bins from ui.R
                                                   Ingrediente
             <- faithful[, 2]
        bins <- seq(min(x), max(x), length.out = input$bins + 1)
        # draw the histogram with the specified number of bins
        hist(x, breaks = bins, col = 'darkgray', border = 'white')
    })
```

Con esto terminamos el análisis del caso base.

## Otros recursos para aprender



#### **Lecturas:**

Mastering shiny (by H. Wickham, en construcción):

https://github.com/hadley/mastering-shiny

**RStudio Shiny Examples** 

https://github.com/rstudio/shiny-examples

**RStudio tutorial** 

https://shiny.rstudio.com/tutorial/

Getting started with shinydashboard

https://rstudio.github.io/shinydashboard/get\_started.html

#### Cursos de Datacamp

- Building Web Applications in R with Shiny: Case Studies
- Building dashboards with shinydashboard

## Otros recursos para aprender



#### Lecturas:

Awesome Shiny: Lista de recursos que sirven para hacer cosas en shiny.

https://github.com/nanxstats/awesome-shiny-extensions

Temas de shiny. Temas pre-programados para trabajar.

https://github.com/rstudio/shinythemes

**RStudio Shiny Gallery** 

https://shiny.rstudio.com/gallery/

Curso profesional en Ciencia de Datos de Platzi. (Contiene un capítulo de programación en Shiny)

https://platzi.com/clases/data/

### Sobre el upload



Para hacer el upload de una aplicación shiny, hay que hacer una cuenta en la siguiente página:

https://www.shinyapps.io >> Sección dashboard.

Aquí se nos proporcionará una llave y un usuario.

Después, hay que configurar esa cuenta en RStudio, en la sección de **Herramientas >> Opciones Globales >> Publicaciones**, y ahí dar de alta una nueva cuenta con las claves anteriores.

Después hay que instalar la librería, con el siguiente comando: install.packages ('rsconnect')

Por último, necesitamos correr un shinyApp que funcione correctamente, y seleccionar el botón "publish". Si es la primera vez que subimos esa aplicación, se nos pedirá un url para publicarlo.

### Sobre el upload



RStudio provee un servicio de paga de servidor para almacenar aplicaciones shiny de manera sencilla (cuyos precios varían desde \$9 a \$299 USD por mes).

Sin embargo, al hacer tu cuenta en RStudio, tienes derecho a publicar hasta un total de 3 aplicaciones (de tamaño moderadamente pequeño) en el servidor de forma gratuita por 24 horas de uso activo.



Nota: En el LNPP contamos con la licencia completa, por si algún miembro del laboratorio requiere hacer una aplicación de este tipo en algún proyecto.

## ¿Qué sigue?



Una propuesta para continuar en el aprendizaje de creación de aplicaciones web shiny efectivas sería el siguiente:

- 1. Aprender conceptos básicos de programación en R. Aprender a hacer loops, condicionales if, else o switch, usar la familia de funciones apply, etc.
- 2. Visualizaciones y visualizaciones interactivas: Librerias ggplot2, plotly, leaflet, DT, Highcharter y otras del conjunto de HTMLWidgets. Igualmente, si uno quiere ser más complejo, aprender a incorporar visualizaciones de D3.js.
- 3. Temas y estructura. Aprender esto nos permite controlar la apariencia de nuestra aplicación y también la manera en cómo acomodar los elementos de nuestra aplicación de manera óptima.
- **4. Programación y diseño web.** Aprender HTML, HTML <tags>, formato con archivos \*.css y nociones de JavaScript.
- 5. Aprender programación reactiva en R. Aprender qué es un evento, y como estos eventos modifican la apariencia, el comportamiento y los procesos dentro de nuestra aplicación.
- 6. Uso de APIs para trabajar con bases de datos externas, modelos de Machine Learning y con usuarios de Python.
- 7. Entender el upload de nuestra aplicación en servidor. Aprender a manejar el dashboard de control de aplicaciones, configurar bien nuestra cuenta, subir a un url de nuestra propiedad, etc. Igualmente, explorar otras alternativas de upload, como Amazon Wb Services.
- 8. Conceptos de UX (User experience).



## Gracias

