

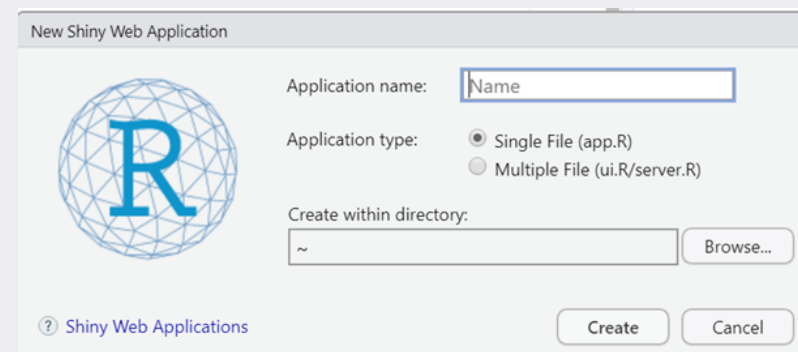
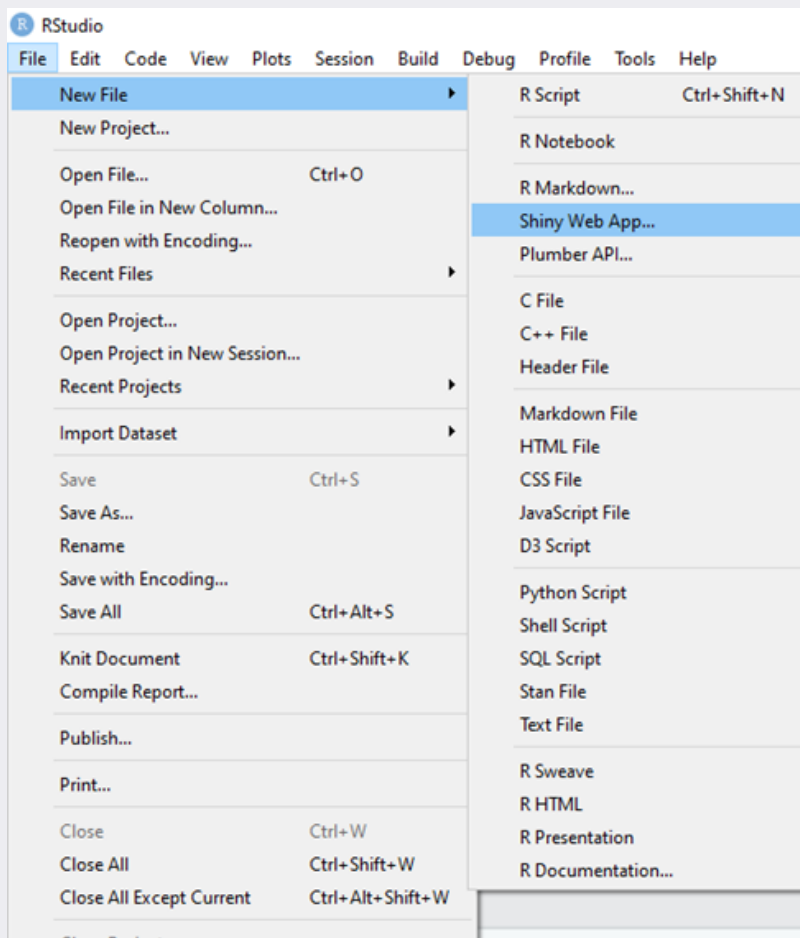
Ayudantía 20 Noviembre

Diplomado en Data Science

Natalie J. - Paula M. - Javiera P.

Shiny ✨

Recordando como crear un archivo shiny:



Librerías que usaremos hoy:

```
library(shiny)           # App web
library(shinydashboard) # Para formato dashboard
library(shinyjs)         # Para usar entorno javascript
library(highcharter)     # Para graficos interactivos
library(DT)              # Para tablas
library(dplyr)           # Para manipulacion de bases de datos
```

Base de datos a utilizar

```
library(readr)
Pokemon <- read_csv("Pokemon.csv")
```

RECOMENDACIÓN

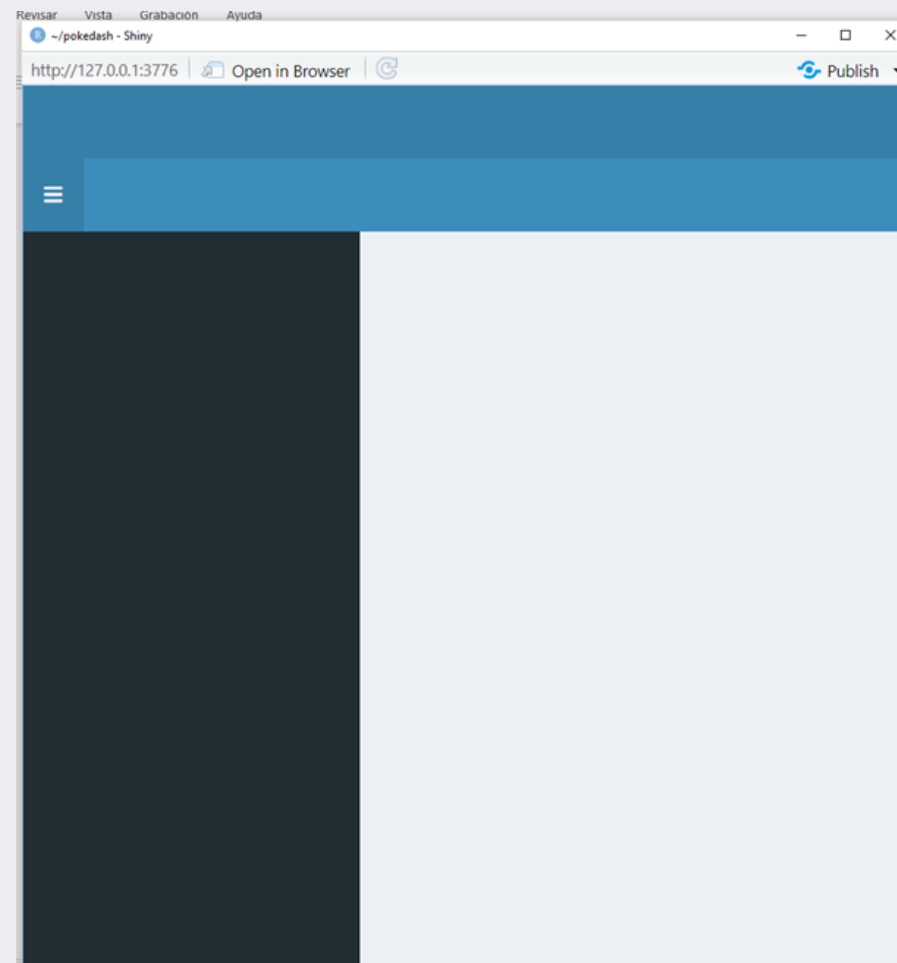
Evita utilizar caracteres con tildes, símbolos y letras "especiales" en tu shiny app, puesto que a veces arroja error de encoding :D

Shiny Dashboard

Creando un Dashboard desde cero:

Estructura:

```
## Barra superior del dashboard:  
header <- dashboardHeader( )  
  
## Menu de navegacion del dashboard:  
sidebar <- dashboardSidebar( )  
  
## Cuerpo de cada vineta del menu  
body <- dashboardBody( )  
  
ui <- dashboardPage(header, sidebar, body)  
server <- function(input, output) {}  
  
shinyApp(ui = ui, server = server)
```



Paquete dashboardtheme

Podemos cambiar el tema predeterminado editando el "Body" de nuestro dashboard

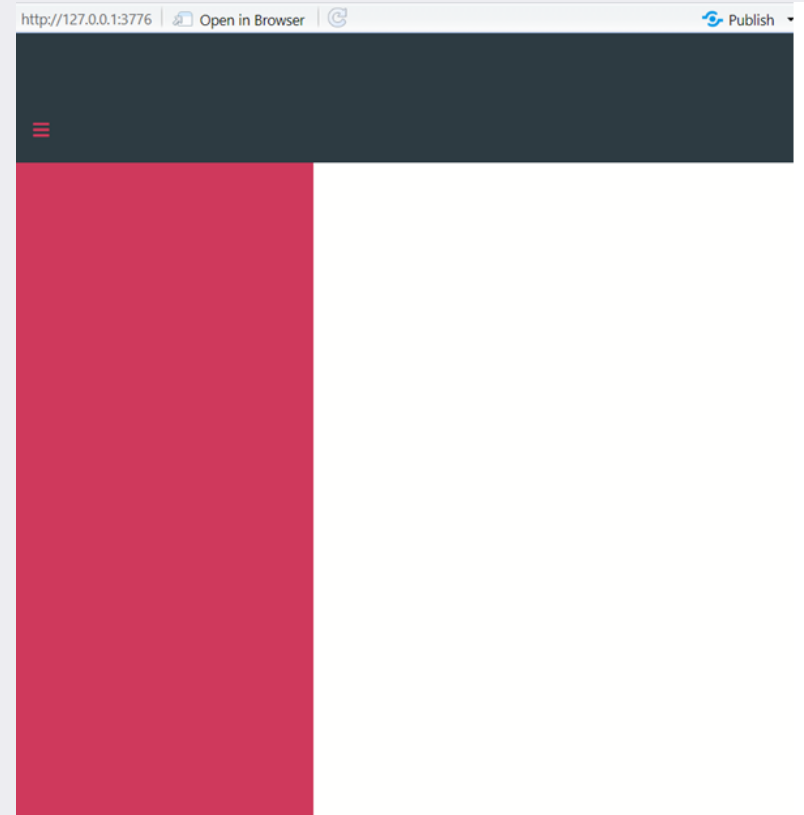
```
## Barra superior del dashboard:
header <- dashboardHeader( )

## Menu de navegacion del dashboard:
sidebar <- dashboardSidebar( )

## Cuerpo de cada vineta del menu
body <- dashboardBody(
  shinyDashboardThemes(
    theme = "flat_red"
  )
)

ui <- dashboardPage(header, sidebar, body)
server <- function(input, output) {}

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



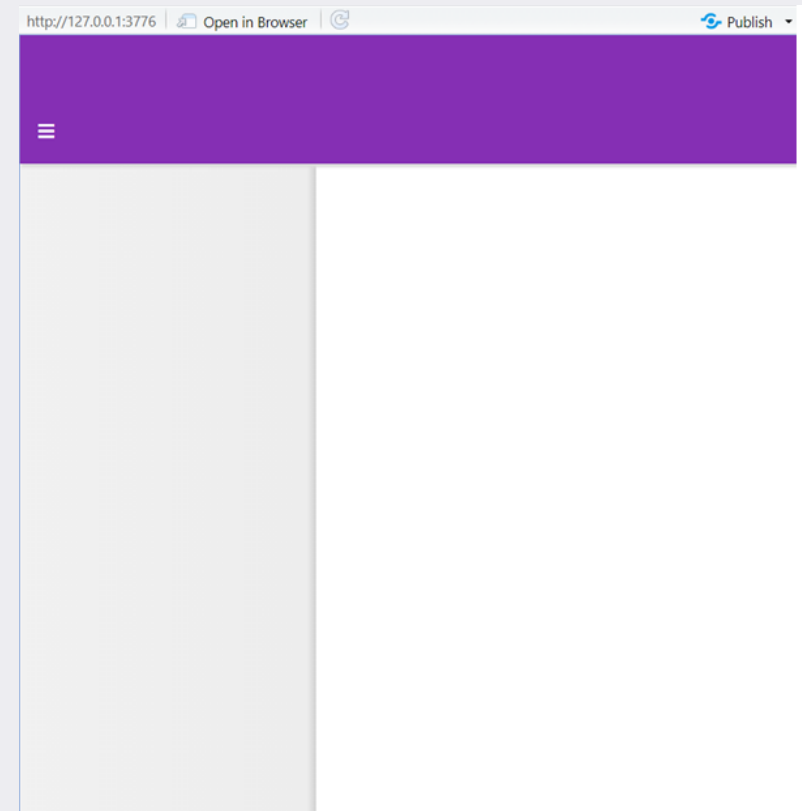
Paquete dashboardtheme

Para revisar más temas ingresar aquí

| Theme | Code | Description |
|-------------------|------------------|--|
| Blue gradient | blue_gradient | Demonstrates use of gradients, shadows and rounded corners |
| Flat Red | flat_red | Flat colour theme with red highlights. |
| Grey light | grey_light | Simple, light theme using a grey colour schemes |
| Grey dark | grey_dark | Demonstrates use of inverted dark colour schemes |
| OneNote | onenote | Styled similarly to the OneNote application |
| Poor man's Flatly | poor_mans_flatly | Poor man's version of the Flatly theme |
| Purple gradient | purple_gradient | Demonstrates high use of gradients and a smaller sidebar |

Cuerpo de cada vineta del menu

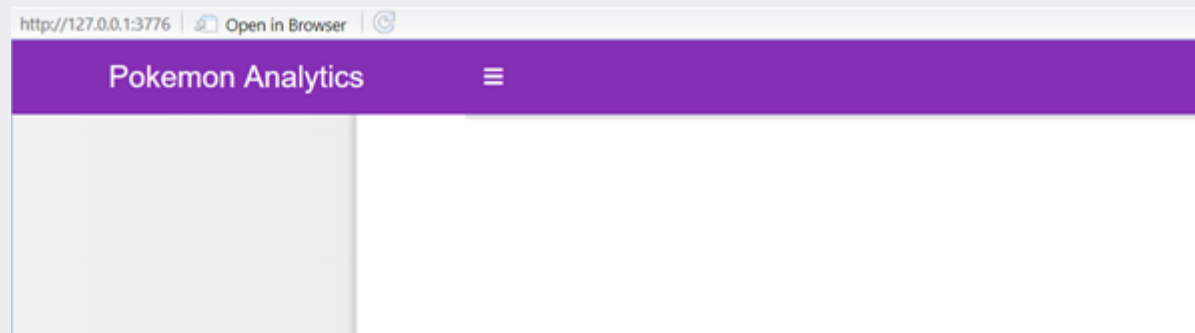
```
body <- dashboardBody(  
  shinyDashboardThemes(  
    theme = "onenote"  
  )  
)
```



Customizando el header

Añadiendo título al dashboard

```
# Barra superior del dashboard:  
header <- dashboardHeader(  
  title="Pokemon Analytics", # Título del dashboard  
  titleWidth=300           # Tamaño del dashboard  
)
```

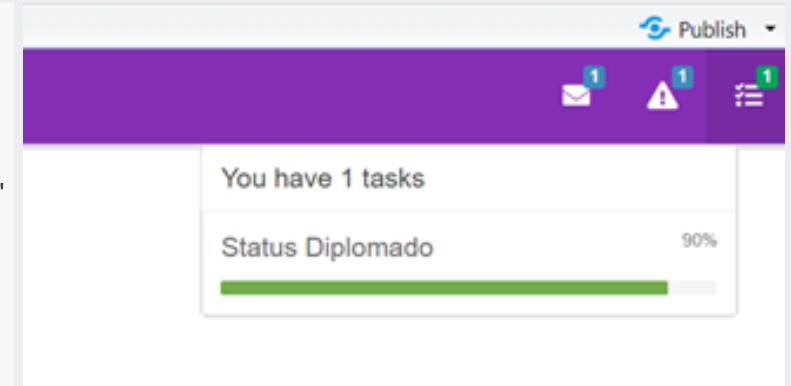


Customizando el header

Añadiendo notificaciones y warnings

```
#Barra superior del dashboard:
header <- dashboardHeader(
  title="Pokemon Analytics",           #Titulo del dashboard
  titleWidth=300,                     #Tamano del dashboard

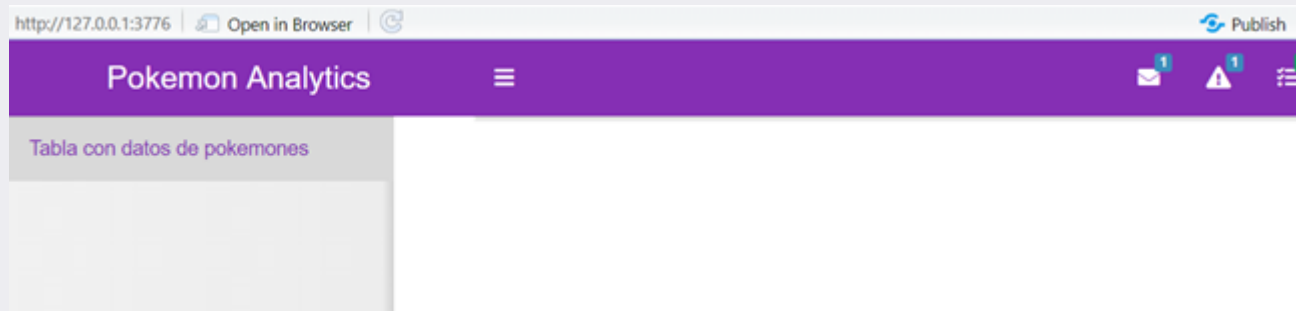
  #Añadiendo notificaciones en el dashboard
  dropdownMenu(type="message",        # Menu emergente del tipo 'mensaje'
    messageItem(
      from = "Las ayudantes dicen:", #'emisor del mensaje'
      message = HTML("¿Dudas? No dudes en consultar :)"), # Mensaje
      icon = icon("question"), #icono del mensaje
      time = substr(Sys.time(), start=12, stop=16)          # Hora que queremos que aparezca
    ),
    dropdownMenu(type = "notifications",
      notificationItem(
        text = "Última ayudantía!!",
        icon("users")
      )
    ),
    dropdownMenu(type = "tasks", badgeStatus = "success",
      taskItem(value = 90, color = "green",
        "Status Diplomado"
      )
    )
  )
```



Customizando el sidebar

Añadiendo una pestaña:

```
# Menu de navegacion del dashboard:
sidebar <- dashboardSidebar(
  width = 250,                                # Tamano del sidebar
  sidebarMenu(
    id='sidebar',                             # Nombre identificador del sidebar
    menuItem('Tabla con datos de pokemones', # Nombre de la pestaña 1 en el dash
              tabName = 'menu1')
  )
)
```

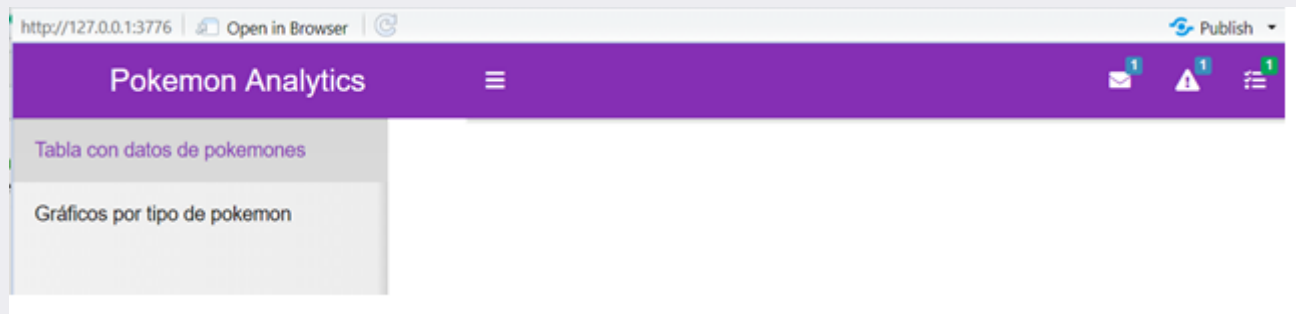


Customizando el sidebar

Añadiendo otra pestaña:

```
# Menu de navegacion del dashboard:
```

```
sidebar <- dashboardSidebar(  
  width = 250,                                # Tamano del sidebar  
  sidebarMenu(  
    id='sidebar',                             # Nombre identificador del sidebar  
    menuItem('Tabla con datos de pokemones',# Nombre de la pestaña 1 en el dash  
              tabName = 'menu1'),  
    menuItem('Graficos por tipo de pokemon',# Nombre de la pestaña 2 en el dash  
              tabName = 'menu2'))  
)
```

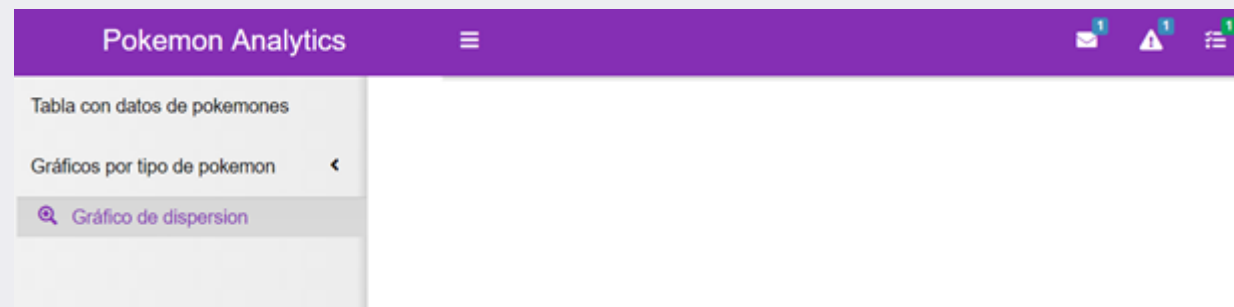


Customizando el sidebar

Añadiendo una pestaña jerárquica:

#Menu de navegacion del dashboard:

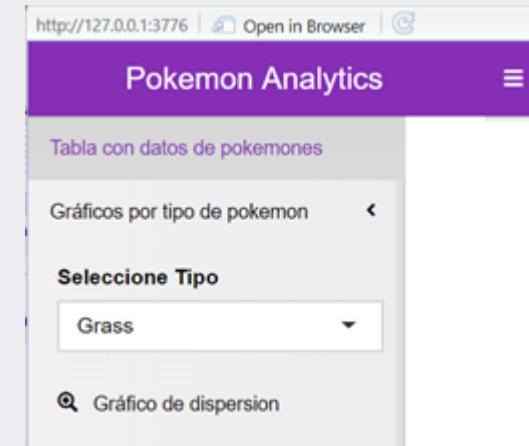
```
sidebar <- dashboardSidebar(  
  width = 250,                                # Tamano del sidebar  
  sidebarMenu(  
    id='sidebar',                             # Nombre identificador del sidebar  
    menuItem('Tabla con datos de pokemones', # Nombre de la pestaña 1 en el dash  
              tabName = 'menu1'),  
    menuItem('Graficos por tipo de pokemon', # Nombre de la pestaña 2 en el dash  
              tabName = 'menu2', startExpanded = T,  
    menuItem('Grafico de dispersion', tabName="menu21",  
              icon = icon("zoom-in", lib = "glyphicon"))  
  )  
)
```



Customizando el sidebar

Añadiendo un input en el sidebar al nivel de la segunda pestaña:

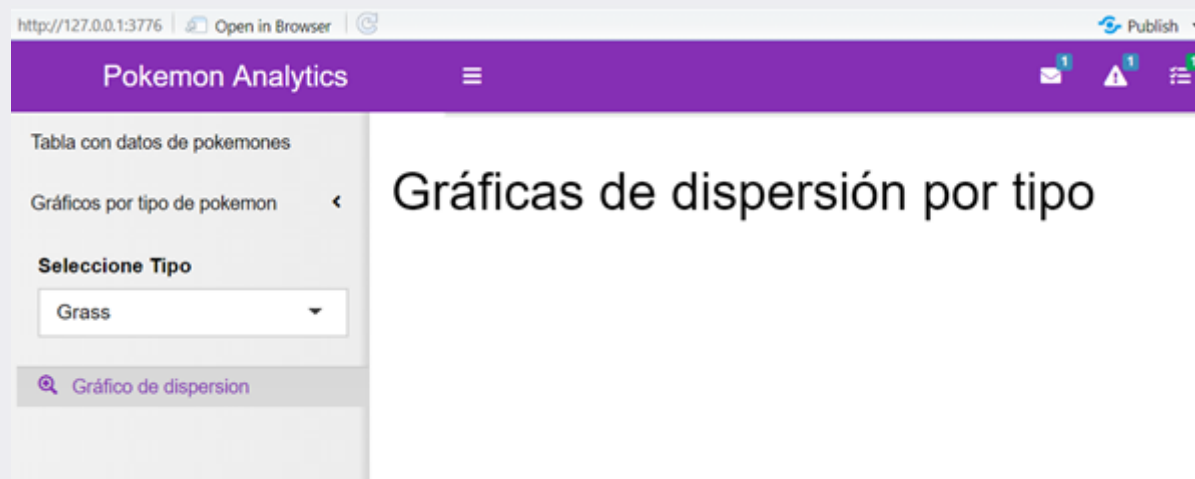
```
#Menu de navegacion del dashboard:
sidebar <- dashboardSidebar(
  width = 250, #Tamano del sidebar
  sidebarMenu(
    id='sidebar', # Nombre identificador del sidebar
    menuItem('Tabla con datos de pokemones', # Nombre de la pestaña 1 en el dash
      tabName = 'menu1'),
    menuItem('Graficos por tipo de pokemon', # Nombre de la pestaña 2 en el dash
      tabName = 'menu2', startExpanded = T,
      div(id = "sidebar1",
        conditionalPanel("input.sidebar === 'menu2'",
          selectizeInput("select_tipo1",
            "Seleccione Tipo",
            choices = unique(Pokemon$`Type 1`),
            selected = "", width = "300px",
            multiple = F))),
      menuItem('Grafico de dispersion', tabName="menu21",
        icon = icon("zoom-in",lib = "glyphicon"))))
  )
)
```



Estructura del body

Estructurando la pestaña "menu21":

```
body <- dashboardBody(  
  shinyDashboardThemes(  
    theme = "onenote"),  
  tabItems(  
    tabItem(tabName = "menu21",  
      h1("Gráficas de dispersión por tipo"),  
      fluidRow(highchartOutput("graf1"))  
    )  
  )  
)
```



Estructura del body (Estructurando la pestaña "menu21")

Definiendo el gráfico dependiendo del input "select_tipo1" en el server.

```
server <- function(input, output) {  
  output$graf1 <- renderHighchart({  
    hchart(Pokemon %>%  
      filter(`Type 1`==input$select_tipo1), # Filtra por el tipo de pokemon seleccionado  
      "scatter", hcaes(x = Speed, y = Attack)) %>% # Grafico de dispersion y variables x e y  
      hc_yAxis(title = list(text = "Attack"))%>% # Titulo eje y  
      hc_title(text=paste("Velocidad y Ataque de Pokemones de tipo",  
        input$select_tipo1), # Titulo del grafico  
        align = "center")%>%  
      hc_tooltip(pointFormat= "Attack: {point.y} <br>  
Speed:{point.x}" ) %>% #tooltip desplegable al posicionar el raton encima de cada punto  
      hc_add_theme(hc_theme_google()) #tema a usar  
  })  
}
```

Estructura del body

Estructurando la pestaña "menu1":

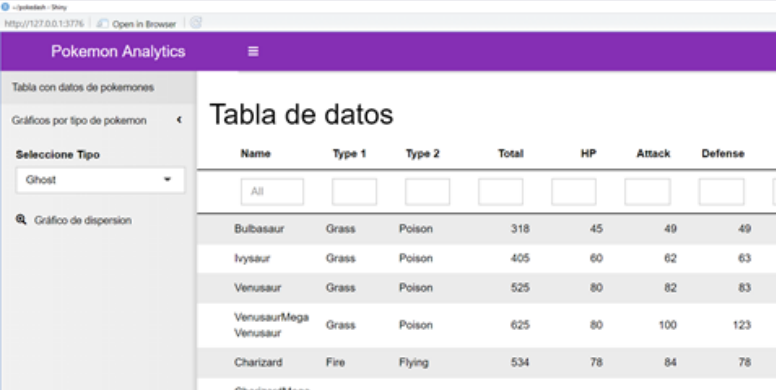
#Cuerpo de cada vineta del menu

```
body <- dashboardBody(  
  shinyDashboardThemes(  
    theme = "onenote"),  
  tabItems(  
    tabItem(tabName = "menu1",  
            h1("Tabla de datos"),  
            fluidRow(dataTableOutput("table1"))  
    ),  
    tabItem(tabName = "menu21",  
            h1("Gráficas de dispersión por tipo"),  
            fluidRow(highchartOutput("graf1"))  
    )  
)
```

Estructura del body (Estructurando la pestaña "menu1")

Definiendo la tabla "table 1" en el server:

```
server <- function(input, output) {  
  output$table1 <- renderDataTable({  
    datatable(na.omit(Pokemon[,-1]), # Datos a mostrar  
      filter = list(position = "top"), # Posicion del buscador  
      options = list(dom="t", # Elimina un search grande de arriba  
        #autoWidth = TRUE, # Esto hace que se ajuste el ancho  
        pageLength = 8, # Se muestran 8 registros por pagina  
        scrollX = TRUE)) # Se avanza con una barra deslizante horizontal  
  })  
  
  output$graf1 <- renderHighchart({  
    hchart(Pokemon %>%  
      filter(`Type 1`==input$select_tipo1), # Filtra por tipo de pokemon seleccionado  
      "scatter", # Grafico de dispersion  
      hcaes(x = Speed, y = Attack)) %>% # variables x e y  
    hc_yAxis(title = list(text = "Attack"))%>% #titulo eje y  
    hc_title(text=paste("Velocidad y Ataque de Pokemones de tipo",  
      input$select_tipo1), # Titulo del grafico  
      align = "center")%>%  
    hc_tooltip(pointFormat= "Attack: {point.y} <br>  
Speed:{point.x}" ) %>% # tooltip desplegable al posicionar el raton encima de cada punto  
    hc_add_theme(hc_theme_google()) #tema a usar  
  })  
}
```



Pokemon Analytics

Tabla con datos de pokemones

Gráficos por tipo de pokemon

Seleccione Tipo

Ghost

Gráfico de dispersion

| Name | Type 1 | Type 2 | Total | HP | Attack | Defense |
|--------------------------|--------|--------|-------|----|--------|---------|
| All | | | | | | |
| Bulbasaur | Grass | Poison | 318 | 45 | 49 | 49 |
| Ivysaur | Grass | Poison | 405 | 60 | 62 | 63 |
| Venusaur | Grass | Poison | 525 | 80 | 82 | 83 |
| VenusaurMega Venusaur | Grass | Poison | 625 | 80 | 100 | 123 |
| Charizard | Fire | Flying | 534 | 78 | 84 | 78 |
| CharizardMega | | | | | | |

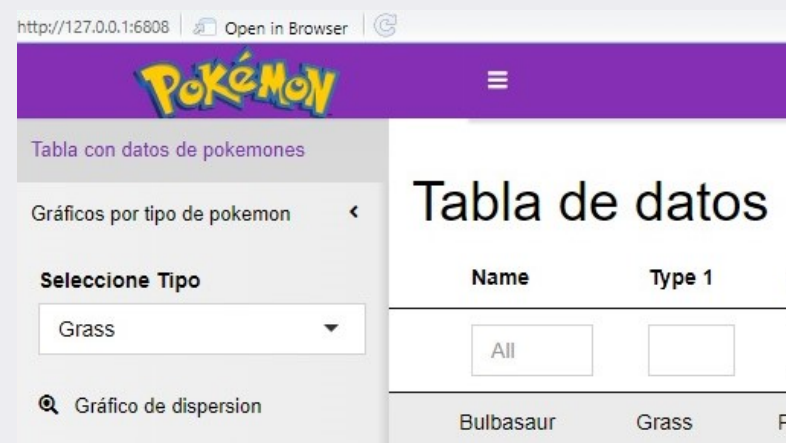
Extra: Añadiendo logo al header

Puede ser de interés añadir un logo a nuestra shiny app. Para ello debemos editar el título con los siguientes comandos:

```
# Barra superior del dashboard:
header <- dashboardHeader(
  title= a(href='https://www.pokemon.com/el/',
           img(src='https://upload.wikimedia.org/wikipedia/commons/9/98/International_Pok%C3%A9mon_logo.svg',
                width='200px',height='50px')),
  ... # Resto del código
```

La personalización usada corresponde a ajustes CSS

- `a()`: Crea un enlace a una página web.
- `img()`: Añade una imagen.



Material complementario:

- <https://shiny.rstudio.com/images/shiny-cheatsheet.pdf>
- <https://shiny.rstudio.com/gallery/>
- <https://jkunst.com/highcharter/articles/shiny.html>

Código final

```
library(shiny) #App web
library(shinydashboard) #Para formato dashboard
library(shinyjs) #Para usar entorno javascript
library(highcharter) #Para graficos interactivos
library(DT) #Para tablas
library(dplyr) #Para manipulacion de bases de datos
library(dashboardthemes) #Para modificar el theme de un shinydashboard

###Base de datos a utilizar
library(readr)
Pokemon <- read_csv("Pokemon.csv")
#Barra superior del dashboard:
header <- dashboardHeader(
  title= a(href='https://www.pokemon.com/el/',
    img(src='https://upload.wikimedia.org/wikipedia/commons/9/98/International_Pok%C3%A9mon_logo.svg',
      width='200px',height='50px')), # Titulo del dashboard con logo
  titleWidth=300, #Tamano del dashboard
  #Anadiendo notificaciones en el dashboard
  dropdownMenu(type="message", #Menu emergente del tipo 'mensaje'
    messageItem(
      from = "Las ayudantes dicen:", #'emisor del mensaje'
      message = HTML("Dudas? No dudes en consultar :)"), #mensaje
      icon = icon("question"), #icono del mensaje
      time = substr(Sys.time(), start=12, stop=16) #hora que queremos que aparezca
    ),
  dropdownMenu(type = "notifications",
    notificationItem(
      text = "Ultima ayudantia!!",
      icon("users")
    ),
  dropdownMenu(type = "tasks", badgeStatus = "success",
    taskItem(value = 90, color = "green",
      "Status Diplomado"
    )
  )
)
#Menu de navegacion del dashboard:
sidebar <- dashboardSidebar(
  width = 250, #Tamano del sidebar
  sidebarMenu(
    id='sidebar', #Nombre identificador del sidebar
    menuItem('Tabla con datos de pokemones', #Nombre de la pestania 1 en el dash
      tabName = 'menu1'),
    menuItem('Graficos por tipo de pokemon', #Nombre de la pestania 2 en el dash
      tabName = 'menu2', startExpanded = T,
      div(id = "sidebar1",
        conditionalPanel("input.sidebar === 'menu2'",
          selectizeInput("select_tipo1",
            "Seleccione Tipo",
            choices = unique(Pokemon$`Type 1`),
            selected = "", width = "300px",
            multiple = F)),
        menuItem('Grafico de dispersion', tabName="menu21",
          icon = icon("zoom-in",lib = "glyphicon"))
      )
    )
  )
)
```

```
#Cuerpo de cada vinieta del menu
body <- dashboardBody(
  shinyDashboardThemes(
    theme = "onenote"),
  tabItems(
    tabItem(tabName = "menu1",
      h1("Tabla de datos"),
      fluidRow(dataTableOutput("table1")))
    ),
    tabItem(tabName = "menu21",
      h1("Graficas de dispersion por tipo"),
      fluidRow(highchartOutput("graf1")))
  )
)
ui <- dashboardPage(header, sidebar, body)
server <- function(input, output) {
  output$table1 <- renderDataTable({
    datatable(na.omit(Pokemon[, -1]), # Datos a mostrar
      filter = list(position = "top"), # Posicion del buscador
      options = list(dom="t", # Elimina un search grande de arriba
        #autoWidth = TRUE, #esto hace que se ajuste el ancho
        pageLength = 8, #Se muestran 8 registros por pagina
        scrollX = TRUE)) # Se avanza con una barra deslizante horizontal
  })

  output$graf1 <- renderHighchart({
    hchart(Pokemon %>%
      filter(`Type 1`==input$select_tipo1), # Filtra por el tipo de pokemon seleccionado
      "scatter", hcaes(x = Speed, y = Attack)) %>% # Grafico de dispersion y variables x e y
    hc_yAxis(title = list(text = "Attack")) %>% #titulo eje y
    hc_title(text=paste("Velocidad y Ataque de Pokemones de tipo",
      input$select_tipo1), # Titulo del grafico
      align = "center") %>%
    hc_tooltip(pointFormat= "Attack: {point.y} <br>
      Speed:{point.x} ") %>% # tooltip desplegable al posicionar el raton encima de cada punto
    hc_add_theme(hc_theme_google()) #tema a usar
  })
}
shinyApp(ui = ui, server = server)
```

¡Gracias!