

Manual FlexDashBoards

Juan Diego & Ricardo Pineda

Septiembre 2020 AD

Explicación de uso de las librerías

Se usaron las siguientes librerías para el modelo. Nótese el uso de *MySQL* para conexión de base de datos.

```
library(flexdashboard)
library(readr)
library(dplyr)
library(DT)
library(RMySQL)
library(pool)
library(xts)
library(lubridate)
library(stringr)
library(parsedate)
library(dygraphs)
```

Luego se hace un pool de conexiones a la Base de datos. Aquí los parámetros pueden cambiar según la dirección de su contenedor, nombre de la base de datos, etc.

```
my_db <- dbPool(
  RMySQL::MySQL(),
  dbname = "parcial1_dp",
  host = "127.0.0.1",
  username = "root"
  #idleTimeout = 20000
  #password = "root"
)
```

Para obtener los datos de una tabla se usa el comando siguiente.

```
df_stats <- as.data.frame(my_db %>% tbl("video_stats"))
df_metadata <- as.data.frame(my_db %>% tbl("video_metadata"))
df_video_data <- as.data.frame(my_db %>% tbl("videos"))
```

Luego se limpian los datos, eliminando los duplicados y cambiando el nombre de las columnas.

```
df_stats <- df_stats[!duplicated(df_stats$id), ]
df_stats <- df_stats %>%
  rename(
    Vistas = viewCount,
    Likes = likeCount,
    'No me gusta' = dislikeCount,
    Favoritos = favoriteCount,
    Comentarios = commentCount
  )
```

```

df_metadata <- df_metadata %>%
  rename(
    id = video_id,
    'Título' = title,
    'Descripción' = description,
    'Código' = iframe,
    Enlace = link
  )
df_metadata <- df_metadata[!duplicated(df_metadata$id), ]
df_stats$id <- substr(df_stats$id, 1, 10)
df_video_data <- df_video_data[!duplicated(df_video_data$content_video_id), ]
df_video_data$content_video_id <- substr(df_video_data$content_video_id, 1, 10)
df_video_data$date <- parse_date(df_video_data$date)
df_video_data <- df_video_data %>%
  rename(
    identificador = id,
    id = content_video_id,
    'Fecha' = date
  )

```

Como los datos están relacionados entre sí, se utiliza el comando *merge* para unir ambas tablas.

```

df_videos <- merge(df_metadata, df_stats, by = "id")
df_videos <- merge(df_videos, df_video_data, by = "id")

```

Hacemos una lista con los números de columnas que querramos usar o presentar en la interfaz visual-

```
cols <- c(1, 2, 5, 6, 7, 8, 9, 10, 14)
```

Estadísticas de Académática

Para hacer las cajas superiores de colores se utilizaron las funciones de FlexDashBoards que respetan la jerarquía del markdown.

```
# Estadísticas de Académática
```

```
## Videos
```

```
### Total de vistas
```

Luego, se usó una *ValueBox* para guardar determinados cálculos.

```
valueBox(sum(df_videos$Vistas), icon = "fa-eye", color = 'primary')
```

Gráfica de línea del tiempo

Para hacer la gráfica de línea del tiempo se usó la librería *dygraph*. Con esta librería se pueden hacer diversas gráficas.

Haciendo un segmento de los datos

Con las funciones de *dplyr* se hizo un nuevo conjunto de datos.

```
df_fecha_likes <- df_videos %>% group_by(as.yearmon(Fecha) ) %>% summarise(total_vistas = sum(Vistas) /
```

Luego se convirtió al tipo *time series*, ya que queremos hacer una línea del tiempo. Se indica bajo qué columna quiere que se ordene.

```
time_series <- xts(df_fecha_likes[, c(2, 3, 4)], order.by = df_fecha_likes$`as.yearmon(Fecha)`)
```

Por último se usa la librería para hacer la gráfica. Los distintos métodos unidos por `%>%` son para añadir detalles o notaciones a la gráfica.

```
dygraph(time_series[, 1]) %>%  
  dyAxis("y", label = "Número de vistas en Millones") %>%  
  dyAxis("x", label = "Fecha de publicación") %>%  
  dySeries("total_vistas", label = "Vistas") %>%  
  dyOptions(stackedGraph = TRUE) %>%  
  dyRangeSelector(height = 20)
```

Tabla de vídeos

Para desplegar la tabla de vídeos usamos la librería *DT*. Agregamos la opción de mostrar filtros para que se pueda buscar datos dentro de esta.

```
df_videos[, cols] %>% DT::datatable(options = list(  
  pageLength = 5  
),  
  rownames = FALSE,  
  filter = 'top'  
) %>% formatDate(9, "toLocaleString")
```

Análisis gráfico mediante *rbokeh*

Usando la librería de *rbokeh* se pueden hacer gráficas interactivas y guardables en el ordenador. Se usó esta librería para hacer gráficas de puntos y de barras.

Gráficas de puntos

Se usa la instrucción *ly_points*.

```
library(rbokeh)  
p <- figure() %>%  
  ly_points(Likes, Comentarios, data = df_videos,  
    hover = list(`Título`, Likes, Comentarios))  
p
```

Gráficas de barras

Se usa la instrucción *ly_hist*.

```
h <- figure(width = 2000, height = 400) %>%  
  ly_hist(Vistas, data = df_videos, breaks = 100) %>%  
  ly_density(Vistas, data = df_videos)  
h
```