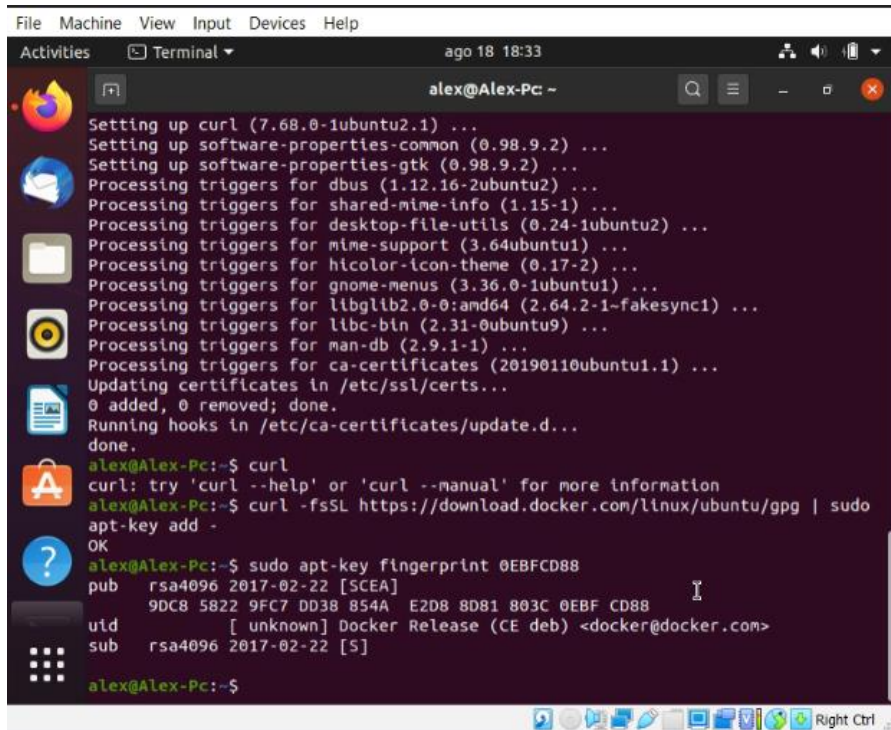


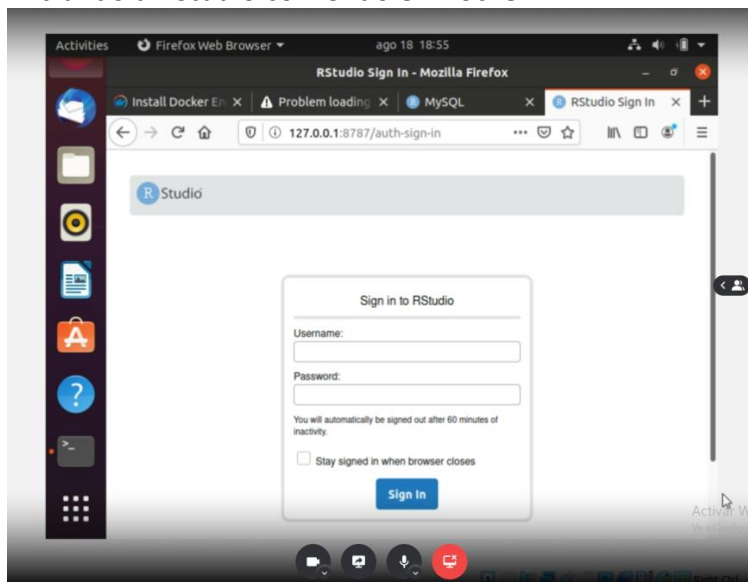
Parcial 1 primera parte

-Instalando Docker

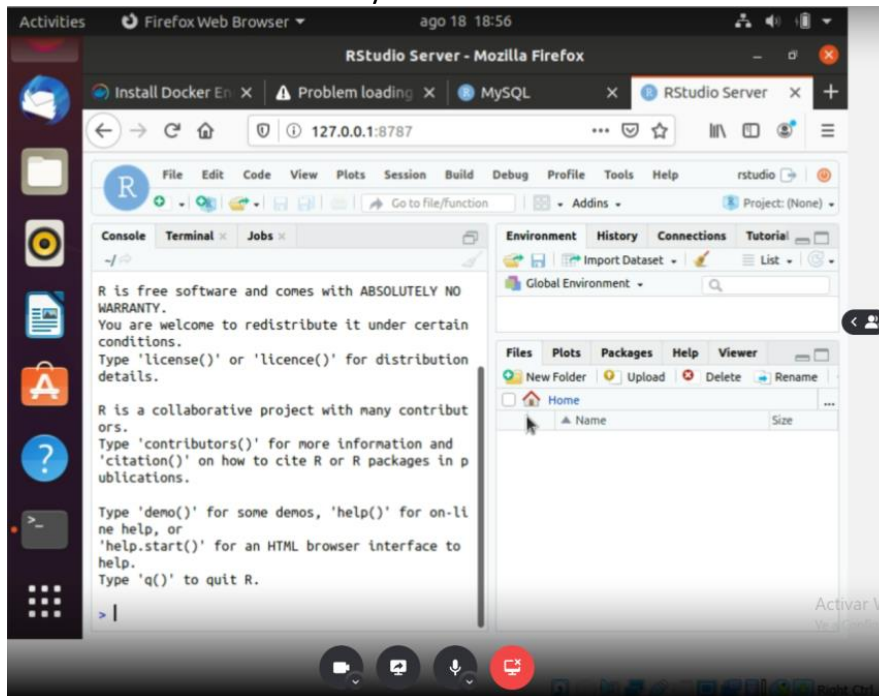


```
File Machine View Input Devices Help
Activities Terminal ago 18 18:33
alex@Alex-Pc: ~
Setting up curl (7.68.0-1ubuntu2.1) ...
Setting up software-properties-common (0.98.9.2) ...
Setting up software-properties-gtk (0.98.9.2) ...
Processing triggers for dbus (1.12.16-2ubuntu2) ...
Processing triggers for shared-mime-info (1.15-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libglib2.0-0:amd64 (2.64.2-1~fakesync1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for ca-certificates (20190110ubuntu1.1) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
alex@Alex-Pc:~$ curl
curl: try 'curl --help' or 'curl --manual' for more information
alex@Alex-Pc:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
apt-key add -
OK
alex@Alex-Pc:~$ sudo apt-key fingerprint 0EBFCD88
pub  rsa4096 2017-02-22 [SCEA]
    9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid      [ unknown] Docker Release (CE deb) <docker@docker.com>
sub  rsa4096 2017-02-22 [S]
alex@Alex-Pc:~$
```

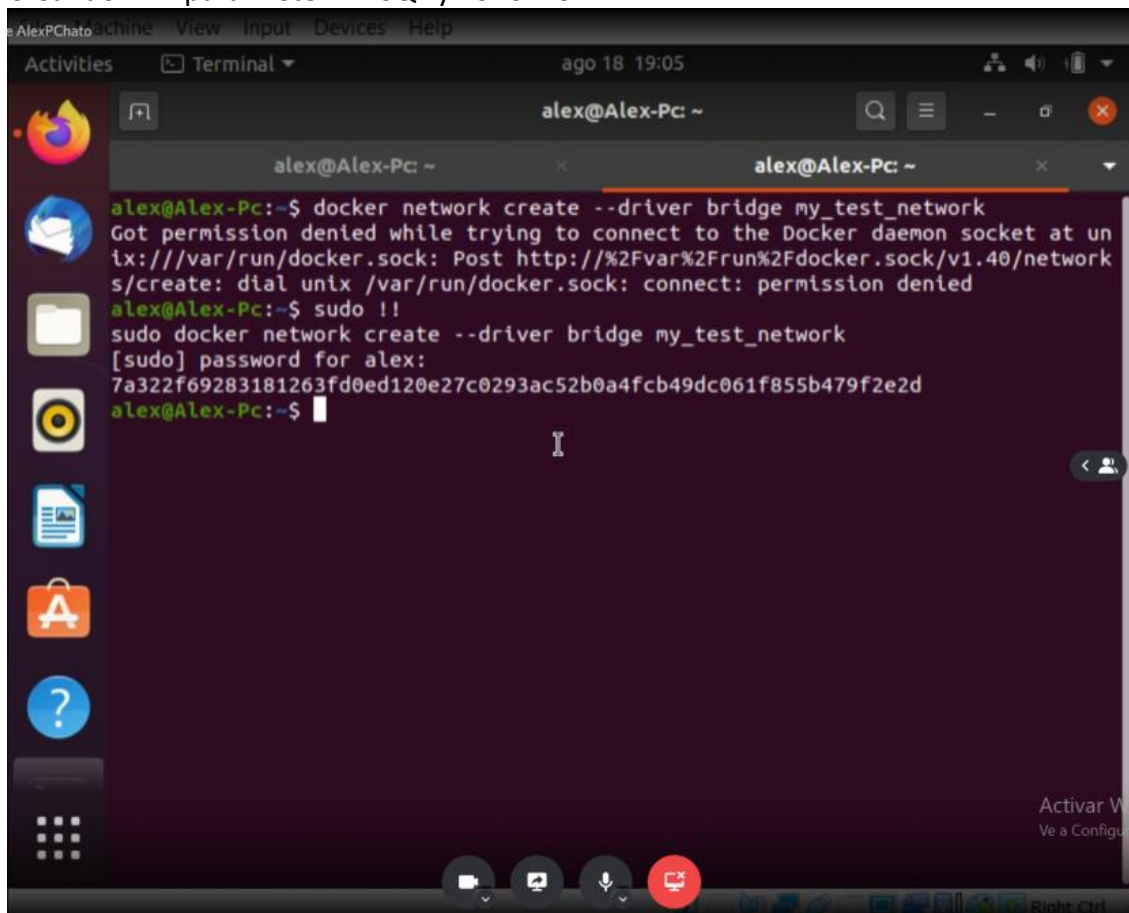
- Entrando a RStudio corriendo en Docker



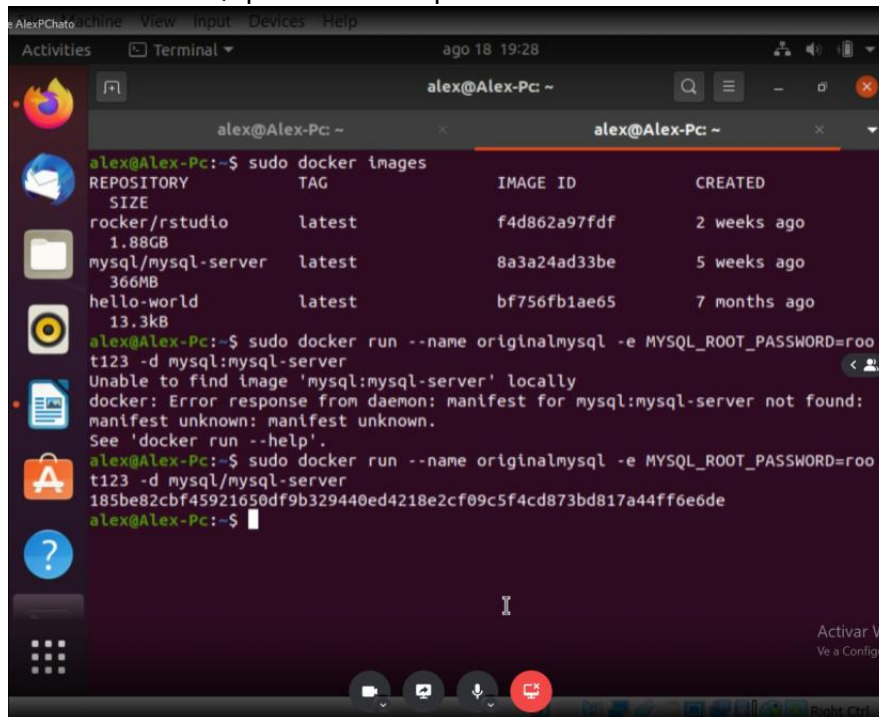
- Acceso a RStudio corriendo ya en el contenedor



- Creando RED para meter MYSQL y RSTUDIO

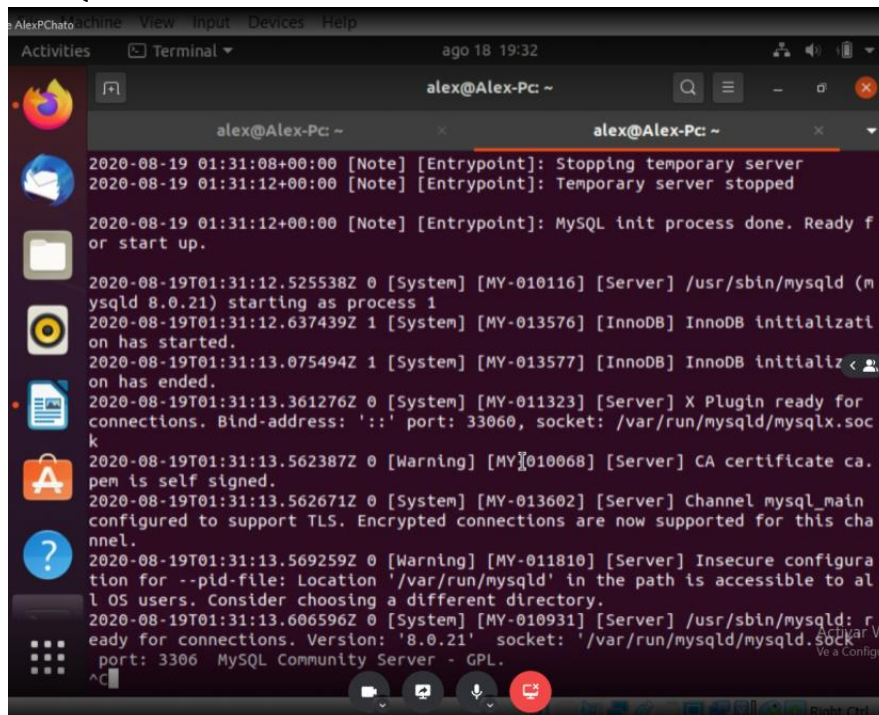


- Corriendo MYSQL para verificar que funciona



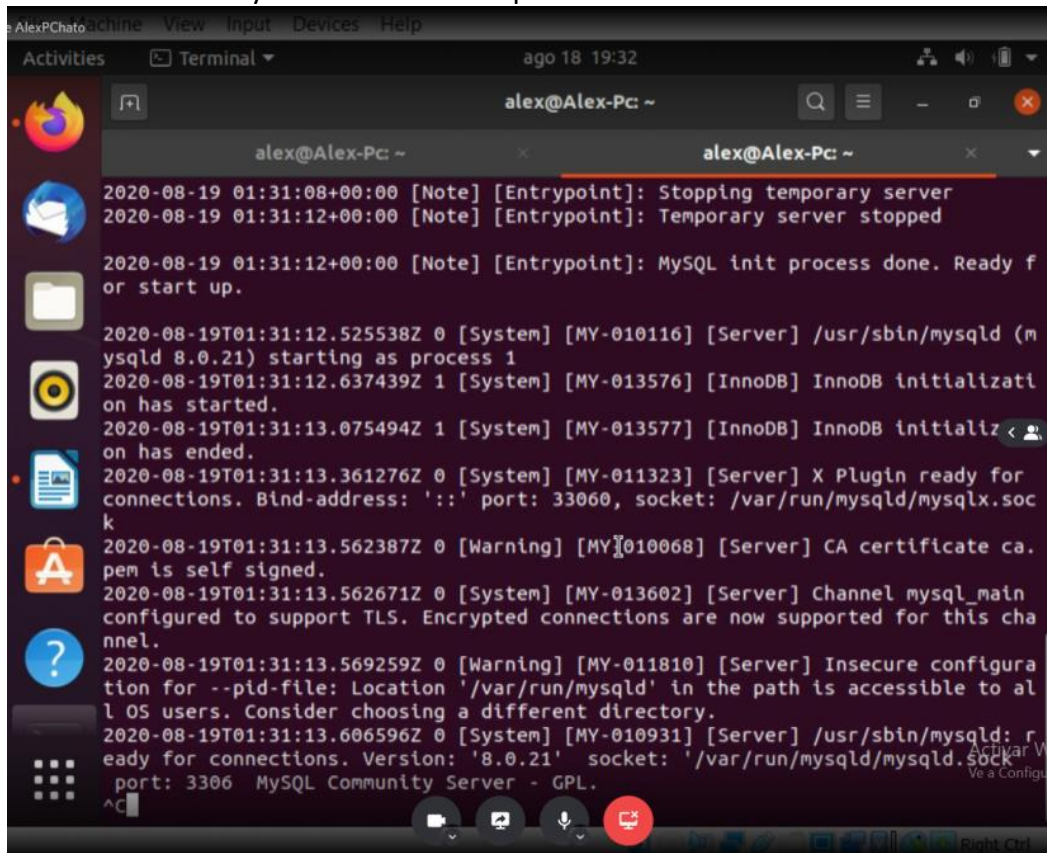
```
AlexPChato@machine View Input Devices Help
ago 18 19:28
alex@Alex-Pc: ~
alex@Alex-Pc: ~
alex@Alex-Pc: ~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
rocker/rstudio       latest             f4d862a97fdf        2 weeks ago
1.88GB
mysql/mysql-server   latest             8a3a24ad33be        5 weeks ago
366MB
hello-world          latest             bf756fb1ae65        7 months ago
13.3kB
alex@Alex-Pc:~$ sudo docker run --name originalmysql -e MYSQL_ROOT_PASSWORD=roo
t123 -d mysql:mysql-server
Unable to find image 'mysql:mysql-server' locally
docker: Error response from daemon: manifest for mysql:mysql-server not found:
manifest unknown: manifest unknown.
See 'docker run --help'.
alex@Alex-Pc:~$ sudo docker run --name originalmysql -e MYSQL_ROOT_PASSWORD=roo
t123 -d mysql/mysql-server
185be82cbf45921650df9b329440ed4218e2cf09c5f4cd873bd817a44ff6e6de
alex@Alex-Pc:~$
```

- MYSQL corriendo



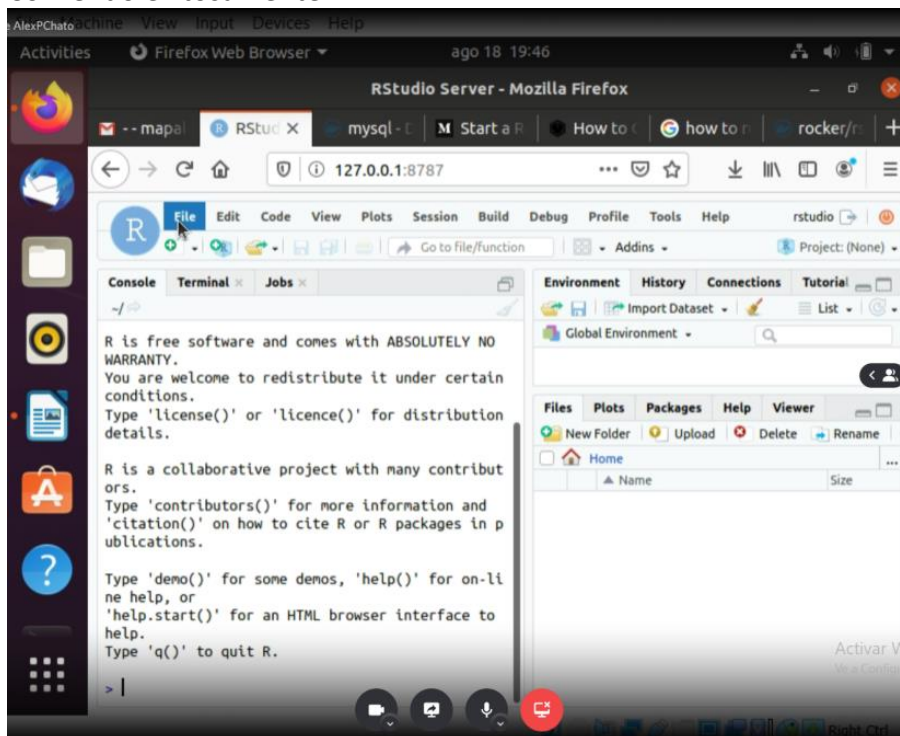
```
AlexPChato@machine View Input Devices Help
ago 18 19:32
alex@Alex-Pc: ~
alex@Alex-Pc: ~
2020-08-19 01:31:08+00:00 [Note] [Entrypoint]: Stopping temporary server
2020-08-19 01:31:12+00:00 [Note] [Entrypoint]: Temporary server stopped
2020-08-19 01:31:12+00:00 [Note] [Entrypoint]: MySQL init process done. Ready f
or start up.
2020-08-19T01:31:12.525538Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (m
ysqld 8.0.21) starting as process 1
2020-08-19T01:31:12.637439Z 1 [System] [MY-013576] [InnoDB] InnoDB initializati
on has started.
2020-08-19T01:31:13.075494Z 1 [System] [MY-013577] [InnoDB] InnoDB initializ
on has ended.
2020-08-19T01:31:13.361276Z 0 [System] [MY-011323] [Server] X Plugin ready for
connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqld.sock
2020-08-19T01:31:13.562387Z 0 [Warning] [MY-010068] [Server] CA certificate ca.
pem is self signed.
2020-08-19T01:31:13.562671Z 0 [System] [MY-013602] [Server] Channel mysql_main
configured to support TLS. Encrypted connections are now supported for this cha
nnel.
2020-08-19T01:31:13.569259Z 0 [Warning] [MY-011810] [Server] Insecure configura
tion for --pid-file: Location '/var/run/mysqld' in the path is accessible to al
l OS users. Consider choosing a different directory.
2020-08-19T01:31:13.606596Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: r
eady for connections. Version: '8.0.21' socket: '/var/run/mysqld/mysqld.sock'
port: 3306 MySQL Community Server - GPL.
^C
```


- Corriendo Rstudio ya dentro de la red que creamos

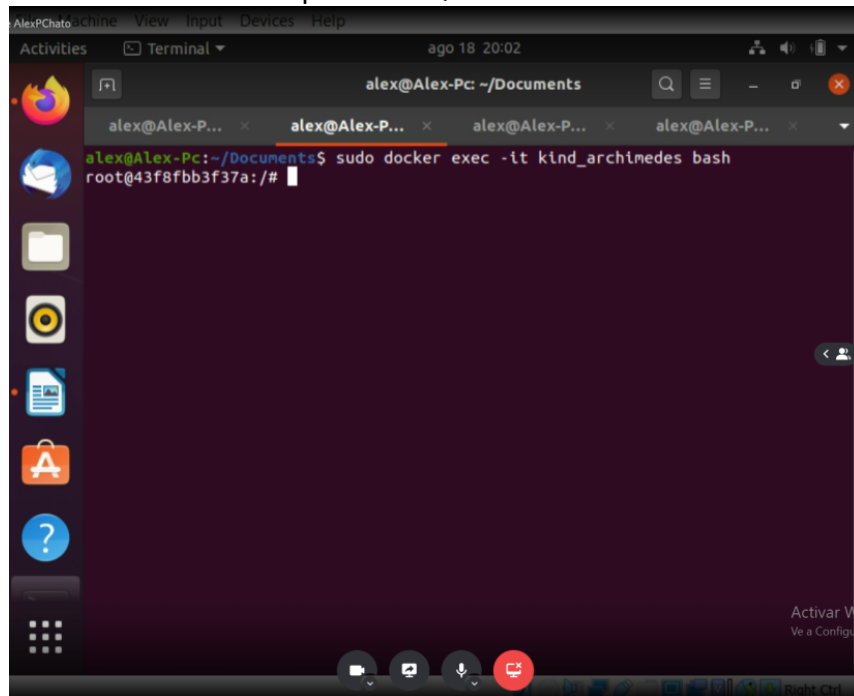


The screenshot shows a terminal window titled 'alex@Alex-Pc: ~' with a dark background. It displays the output of a MySQL service startup. The logs include timestamps, log levels (Note, System, Warning), and component names (Entrypoint, MySQL, InnoDB, X Plugin). Key messages include 'Stopping temporary server', 'Temporary server stopped', 'MySQL init process done. Ready for start up.', 'starting as process 1', 'InnoDB initialization has started.', 'InnoDB initialization has ended.', 'X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqld.sock', 'CA certificate ca.pem is self signed.', 'Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.', 'Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.', and 'ready for connections. Version: '8.0.21' socket: '/var/run/mysqld/mysqld.sock' port: 3306 MySQL Community Server - GPL.'

- Corriendo exitosamente

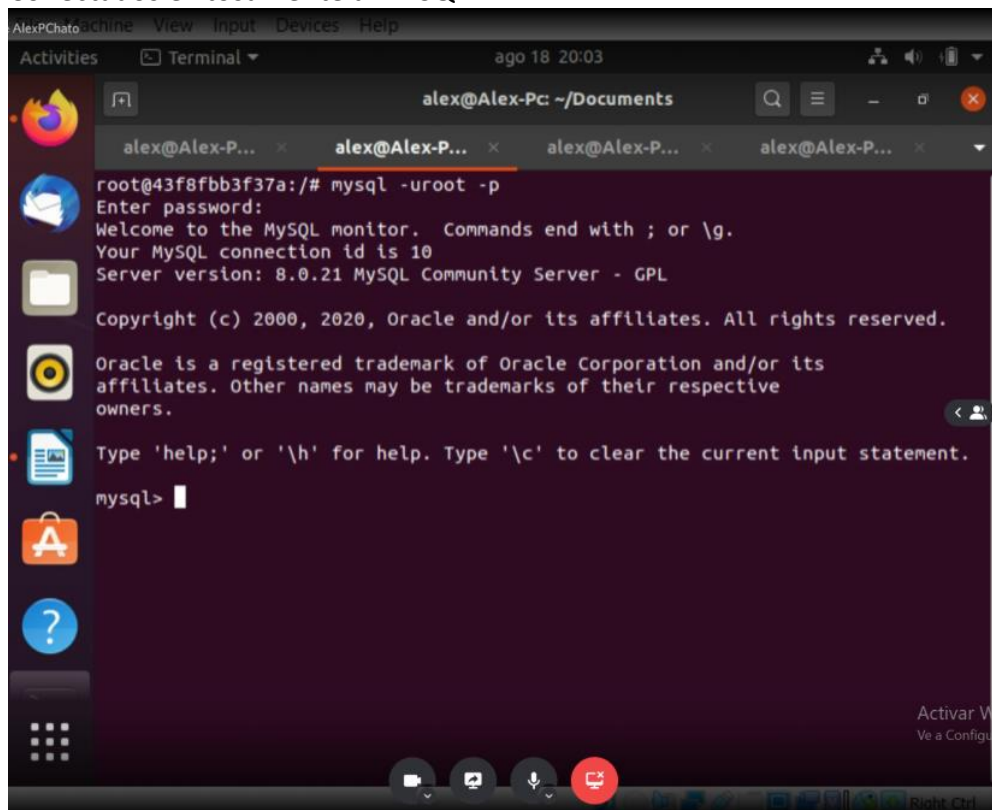


- Abriendo bash para MYSQL



```
alex@Alex-Pc: ~/Documents$ sudo docker exec -it kind_archimedes bash
root@43f8fbb3f37a: /#
```

- Conectados exitosamente a MYSQL



```
root@43f8fbb3f37a: /# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.21 MySQL Community Server - GPL

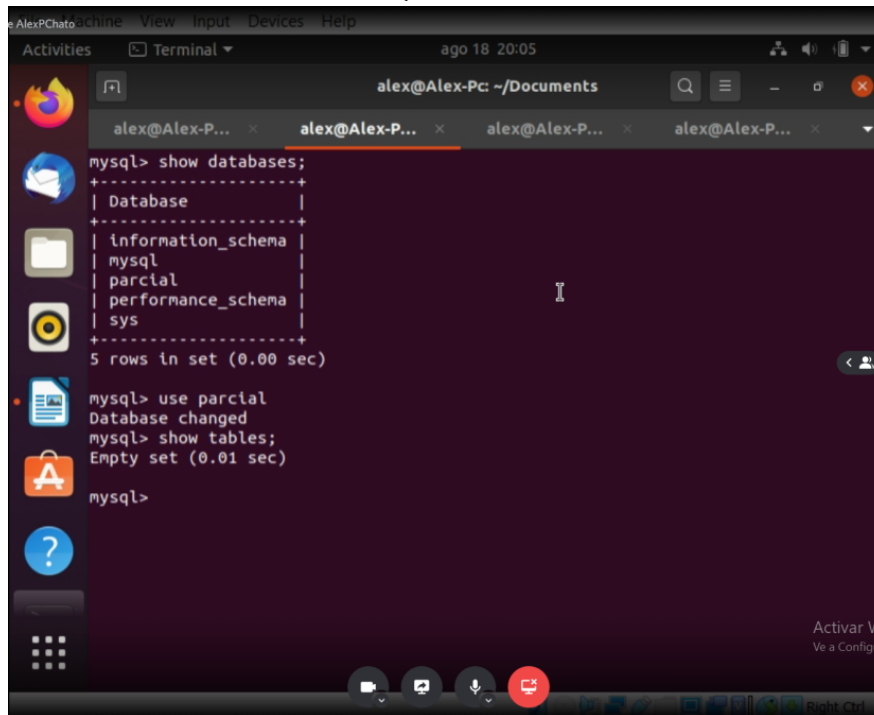
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

- Revisando contenido de MYSQL



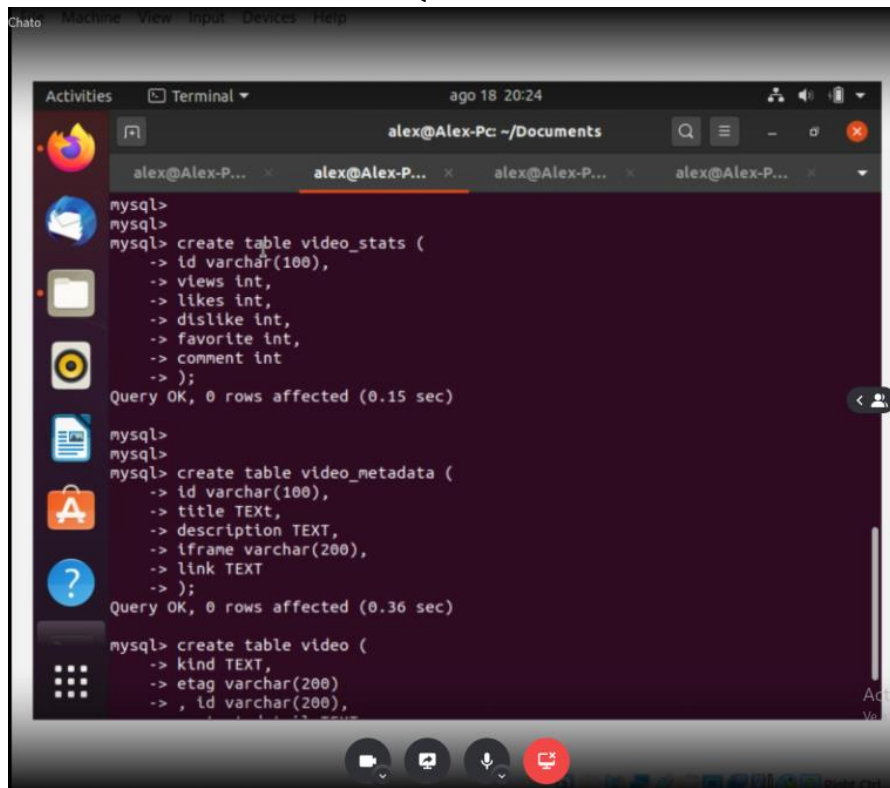
A screenshot of a terminal window titled 'alex@Alex-Pc: ~/Documents'. The terminal shows the following commands and output:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| partial |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> use partial
Database changed
mysql> show tables;
Empty set (0.01 sec)

mysql>
```

- Creando tablas dentro de MYSQL

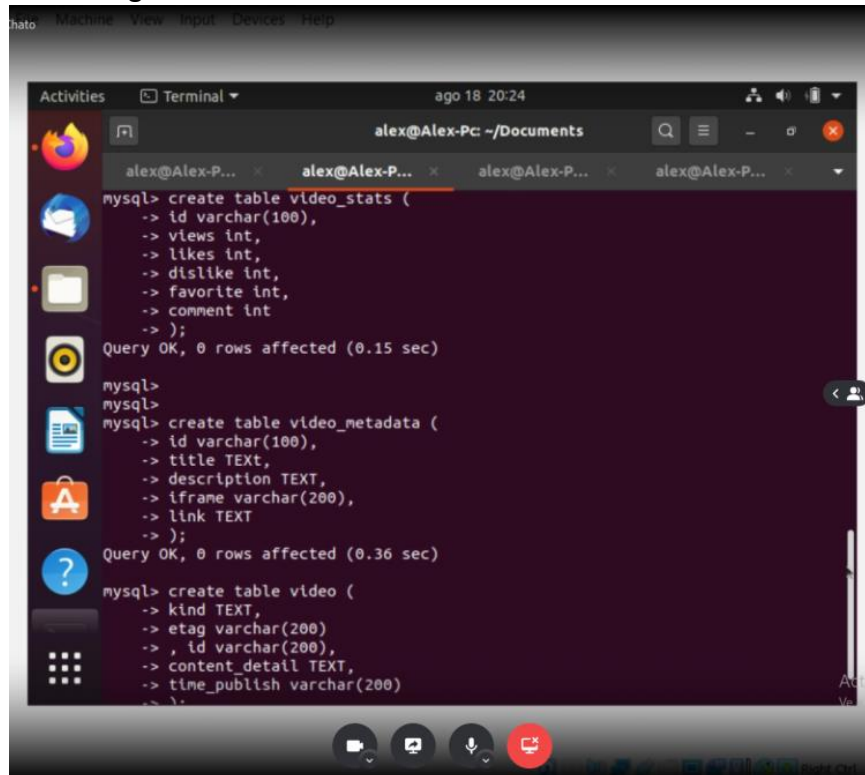


A screenshot of a terminal window titled 'alex@Alex-Pc: ~/Documents'. The terminal shows the following commands and output:

```
mysql>
mysql> create table video_stats (
-> id varchar(100),
-> views int,
-> likes int,
-> dislike int,
-> favorite int,
-> comment int
-> );
Query OK, 0 rows affected (0.15 sec)

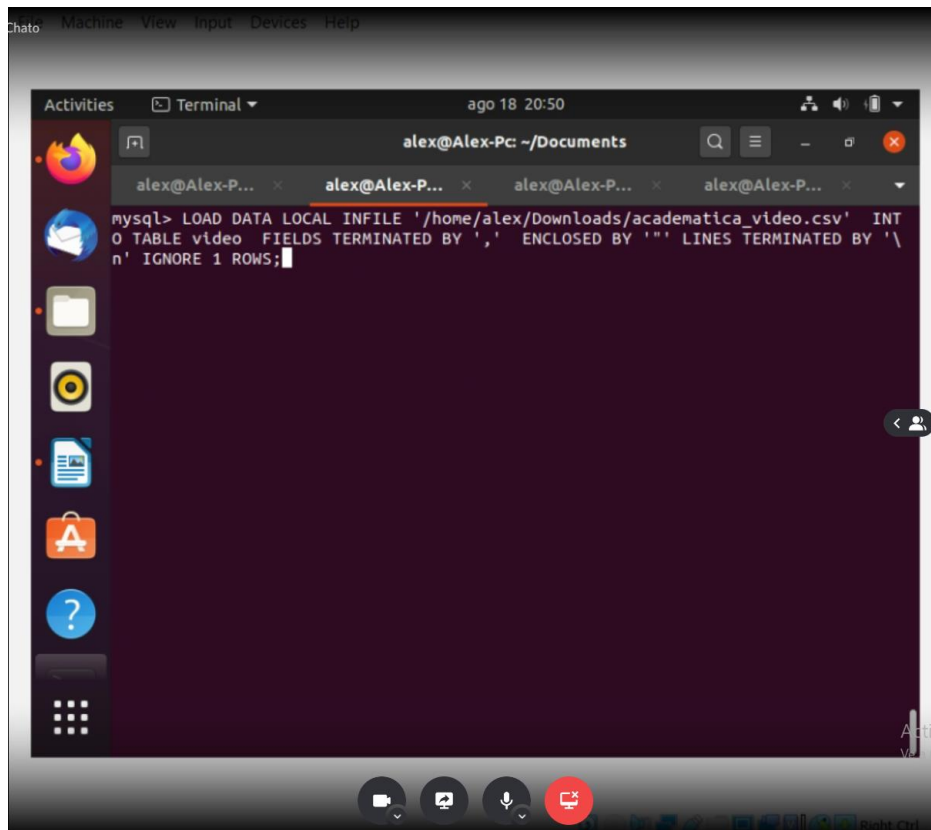
mysql>
mysql> create table video_metadata (
-> id varchar(100),
-> title TEXT,
-> description TEXT,
-> iframe varchar(200),
-> link TEXT
-> );
Query OK, 0 rows affected (0.36 sec)

mysql> create table video (
-> kind TEXT,
-> etag varchar(200)
-> , id varchar(200),
```



```
mysql> create table video_stats (  
-> id varchar(100),  
-> views int,  
-> likes int,  
-> dislike int,  
-> favorite int,  
-> comment int  
-> );  
Query OK, 0 rows affected (0.15 sec)  
  
mysql>  
mysql> create table video_metadata (  
-> id varchar(100),  
-> title TEXT,  
-> description TEXT,  
-> iframe varchar(200),  
-> link TEXT  
-> );  
Query OK, 0 rows affected (0.36 sec)  
  
mysql> create table video (  
-> kind TEXT,  
-> etag varchar(200)  
-> , id varchar(200),  
-> content_detail TEXT,  
-> time_publish varchar(200)  
-> );
```

- LOADING DATA TO TABLES

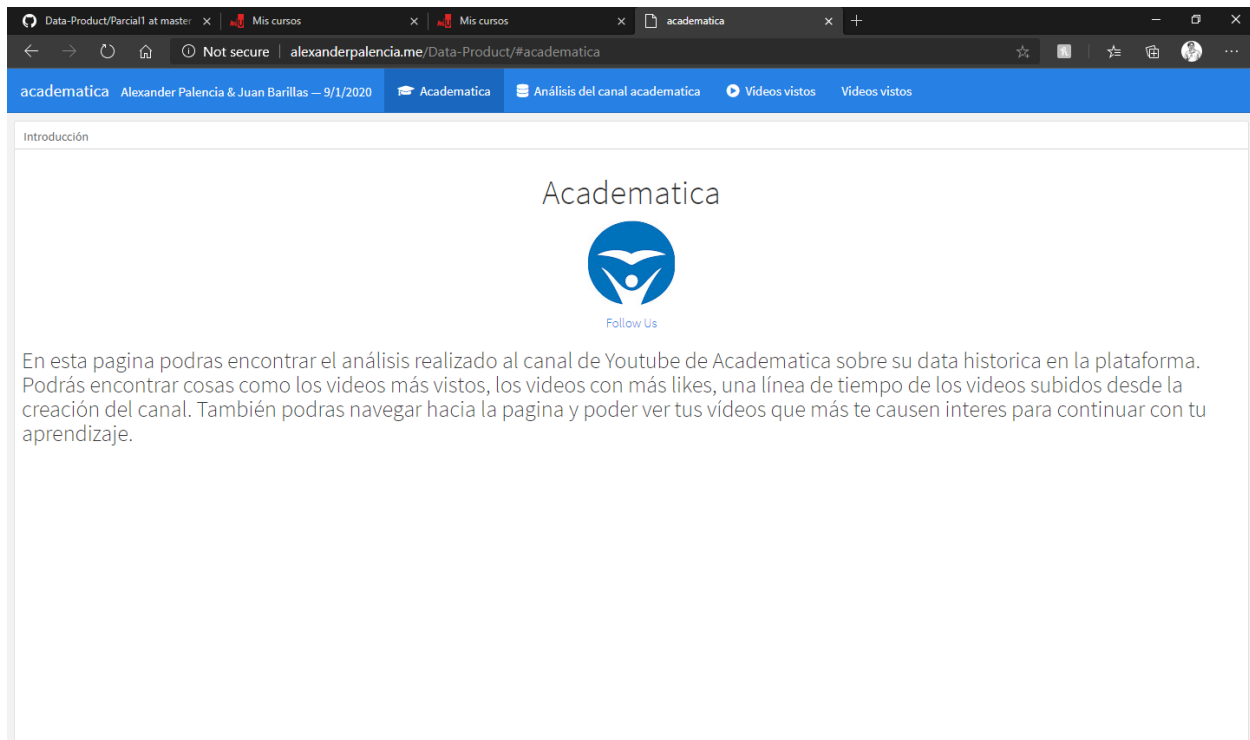


```
mysql> LOAD DATA LOCAL INFILE '/home/alex/Downloads/academática_video.csv' INTO  
TABLE video FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n'  
IGNORE 1 ROWS;
```

Parcial 1 Fase 2

Para la entrega de la fase 2 del proyecto queríamos que nuestro trabajo pudiera ser visualizado de una manera más simple por lo que subimos nuestro html final hacia un servidor de uno de los integrantes por lo que pueden consultarlo en el siguiente link:

<http://alexanderpalencia.me/Data-Product/#videos-vistos>



- Empezamos el trabajo de manera local para poder hacer nuestras pruebas y luego poder hacer un deploy.


```
1 # ---
2 title: "academática"
3 author: "Alexander Palencia & Juan Barillas"
4 date: "9/1/2020"
5 output:
6   flexdashboard::flex_dashboard:
7     orientation: rows
8     vertical_layout: fill
9 # ---
10
11 # {r librerías}
12 library(flexdashboard)
13 library(readr)
14 library(dplyr)
15 library(purrr)
16 library(tidyverse, warn.conflicts = FALSE)
17 library(lubridate, warn.conflicts = FALSE)
18 library(plotly)
19 library(ggplot2)
20 # ---
21
22 # {r carga archivo}
23 stats <- read_csv('data/academática_video_stats.csv')
24 metadata <- read_csv('data/new_metadata.csv')
25 videos <- read_csv('data/new_videos.csv')
26
27 metadata <- metadata %>%
28   rename(id = video_id)
29 union <- merge(stats, metadata, all = TRUE, by = "id") %>%
30   arrange(id)
31 df = subset(videos, select = c(contentDetails.videoId, date)) %>%
32   rename(id = contentDetails.videoId) %>%
33   arrange(id)
34
35 finalTable <- merge(union, df, all = TRUE, by = "id")
36 finalTable <- na.omit(finalTable)
37
38 videosYear <- year(ymd(finalTable$date))
39 videosMonth <- month(ymd(finalTable$date))
40 videosDay <- day(ymd(finalTable$date))
41
42 finalTable$yearPublish <- videosYear
43 finalTable$monthPublish <- videosMonth
44 finalTable$dayPublish <- videosDay
45 # ---
```

- En esta parte se evidencia la creación de las distintas variables y dataframes que nos permitirían analizar la información.
- Se creo una tabla unificada para poder trabajar de una manera más dinámica

```
# Academática {data-icon="fa-graduation-cap"}

### Introducción
<div>
  <center>
    <h1>Academática</h1>
    
    <br>
    <h5>
      <a href="https://www.youtube.com/user/chzelada">Follow Us</a>
    </h5>
  </center>
  <h3>
    En esta pagina podras encontrar el análisis realizado al canal de Youtube de Academática sobre su data historica en la
    plataforma. Podrás encontrar cosas como los videos más vistos, los videos con más likes, una línea de tiempo de los videos
    subidos desde la creación del canal. También podras navegar hacia la pagina y poder ver tus videos que más te causen interes
    para continuar con tu aprendizaje.
  </h3>
</div>

# Análisis del canal academática {data-icon="fa-database"}

##
### Estadísticas básicas
{r}
valueBox(nrow(videos),
  caption = "Total de videos publicados",
  icon = 'fa-play-circle',
  color = '#E1E289')
...

###
{r}
valueBox(nrow(stats),
  caption = "Total de videos con Data",
  icon = 'fa-database',
  color = '#14591D')
...

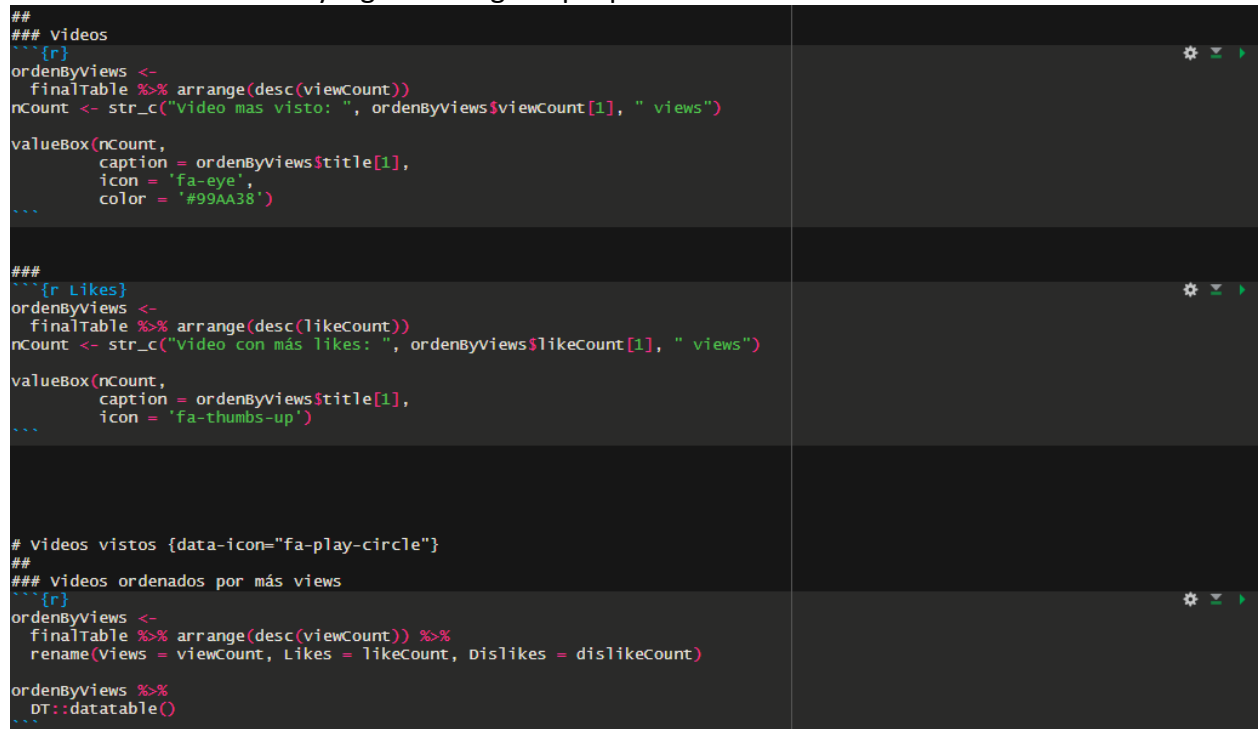
```

- En esta imagen se puede evidenciar la creación de la página principal de nuestro Flexdashboard en el cuál damos una pequeña introducción de datos que se podrán

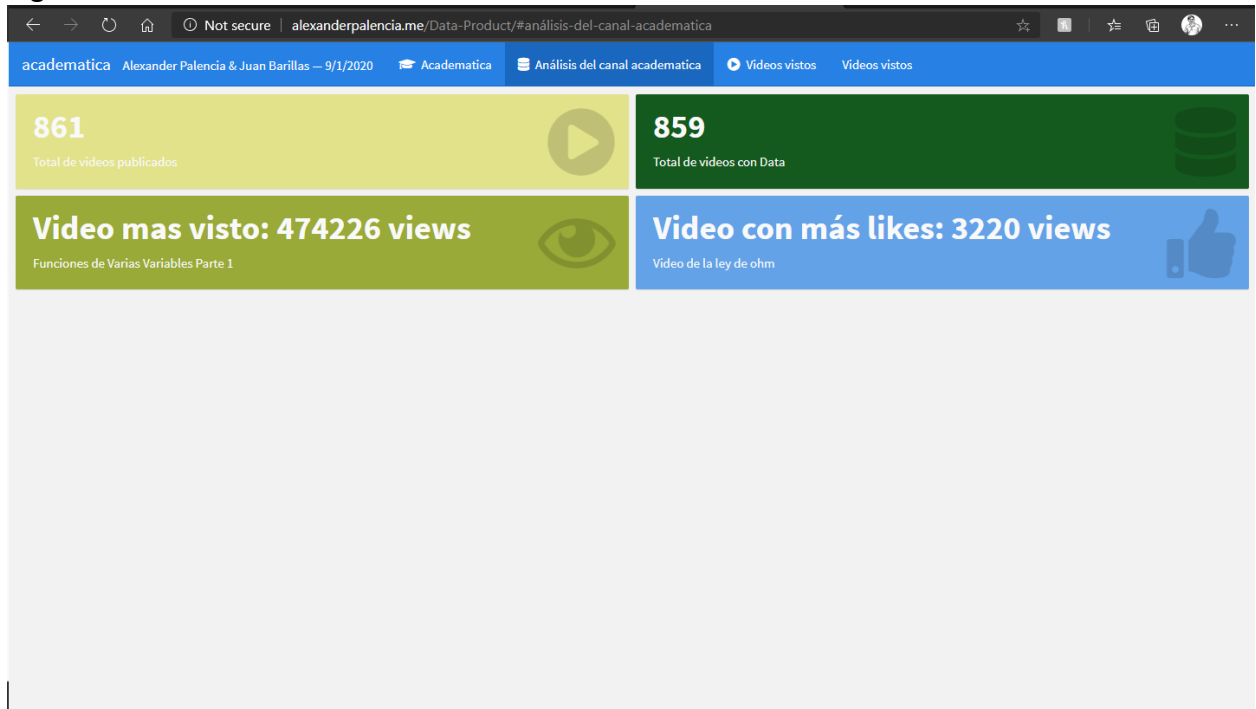
encontrar en la pagina subida al servidor. El output de este código se muestra a continuación.



-
- Luego decidimos presentar información interesante que pudimos encontrar sobre los videos de Academática y algunos insights que pudimos identificar.



-
- El output de este código se muestra en la siguiente imagen



-
- Luego quisimos utilizar algunas de las técnicas aprendidas en clase por lo cuál utilizamos de una manera dinámica el dashboard.

```
###
```{r}

ordenByviews <-
 finalTable %>% arrange(desc(viewCount))
names <- ordenByviews$title[1:5]
variable <- ordenByviews$viewCount[1:5]
fig <- plot_ly(
 x = names,
 y = variable,
 name = "SF Zoo",
 type = "bar"
)
fig

videos_vistos {.storyboard}
=====

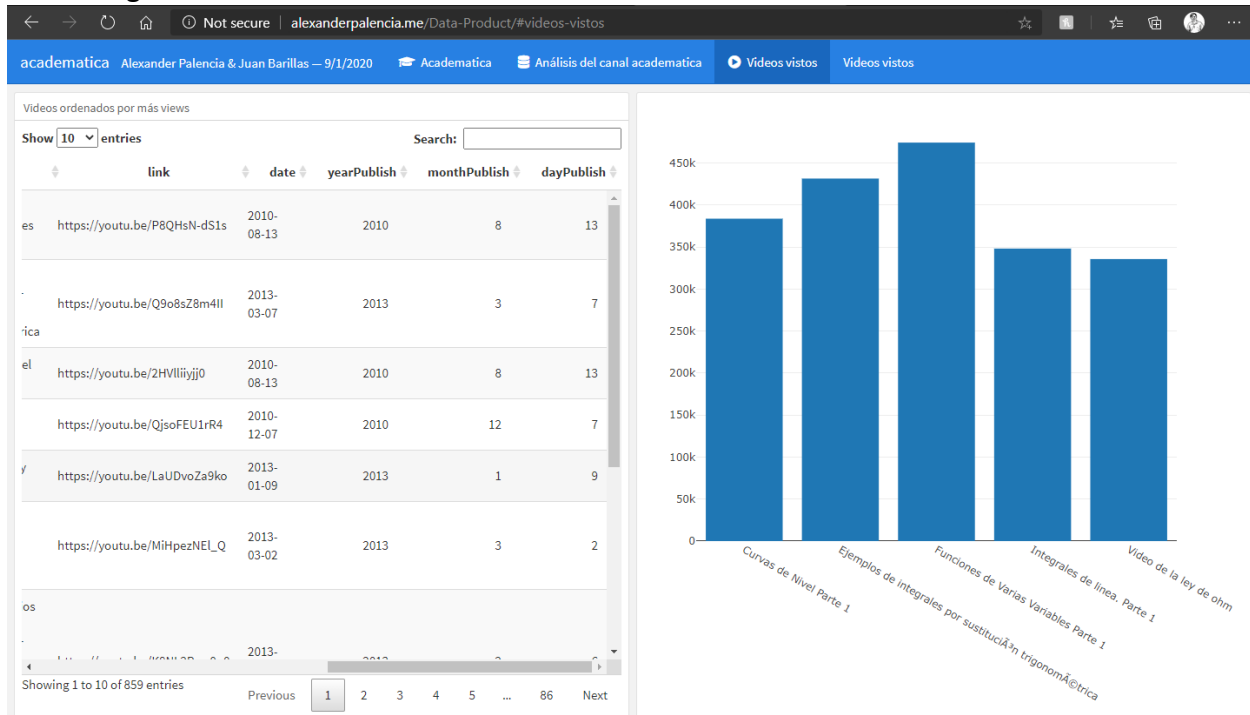
Likes by year

```{r}
x <- ggplot(FinalTable, aes(x = yearPublish, y = likecount, color = yearPublish)) + geom_jitter()
x
```

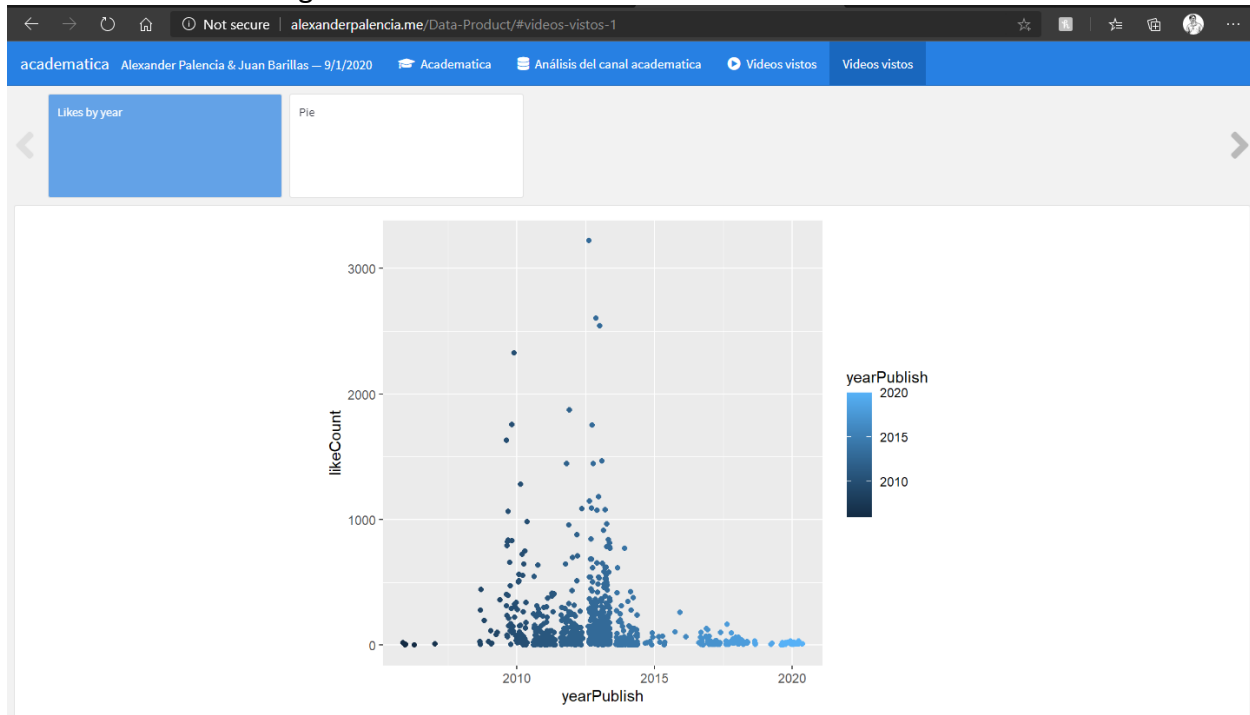
pie

```{r}
slices <- c(sum(finalTable$likeCount), sum(finalTable$dislikeCount))
lbls <- c("Likes", "Dislikes")
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices,labels = lbls, main="Likes vs Dislikes")
```
```

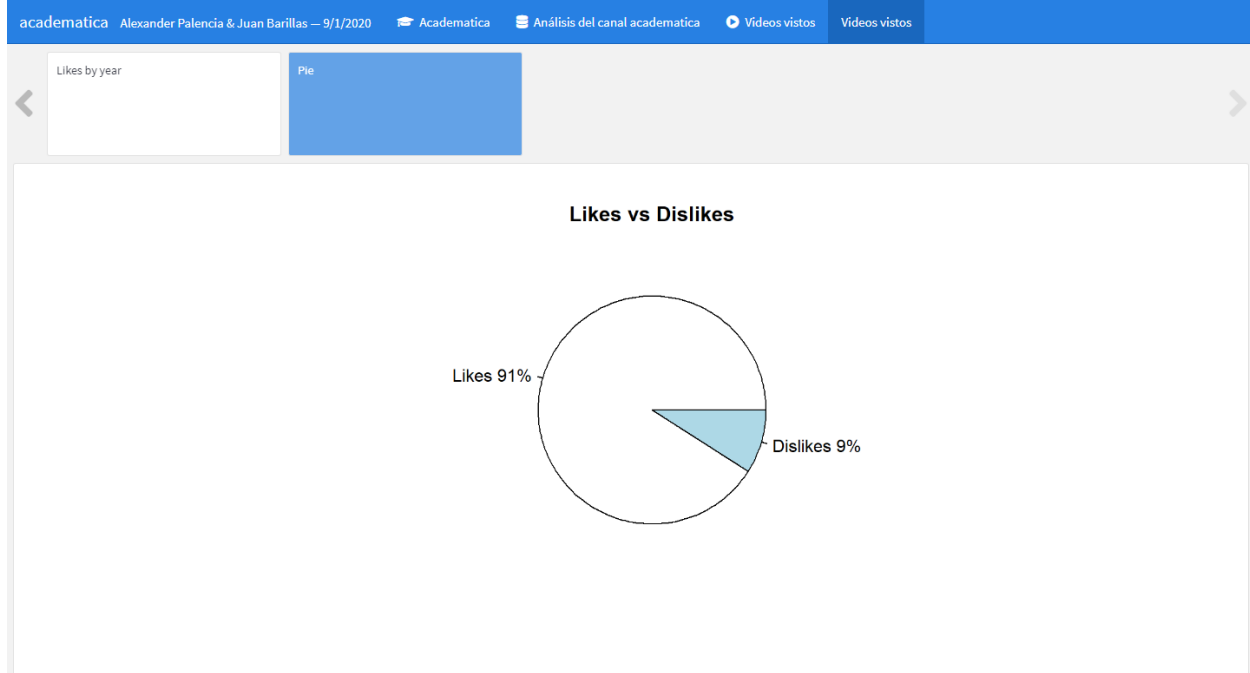
Este código se transforma a la siguiente visualización



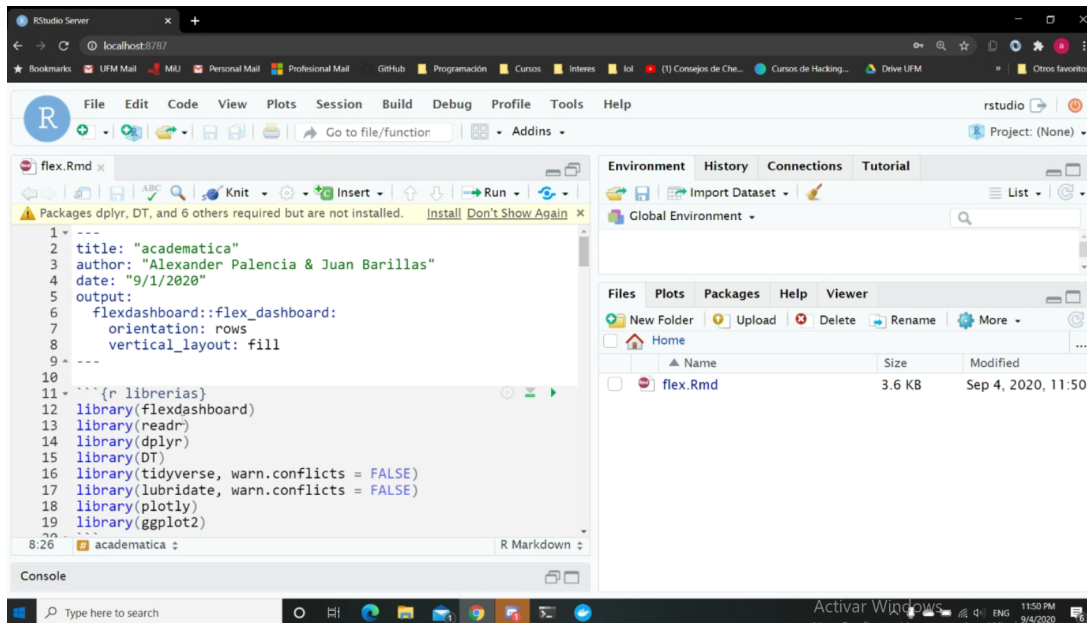
Así también como la siguiente visualización

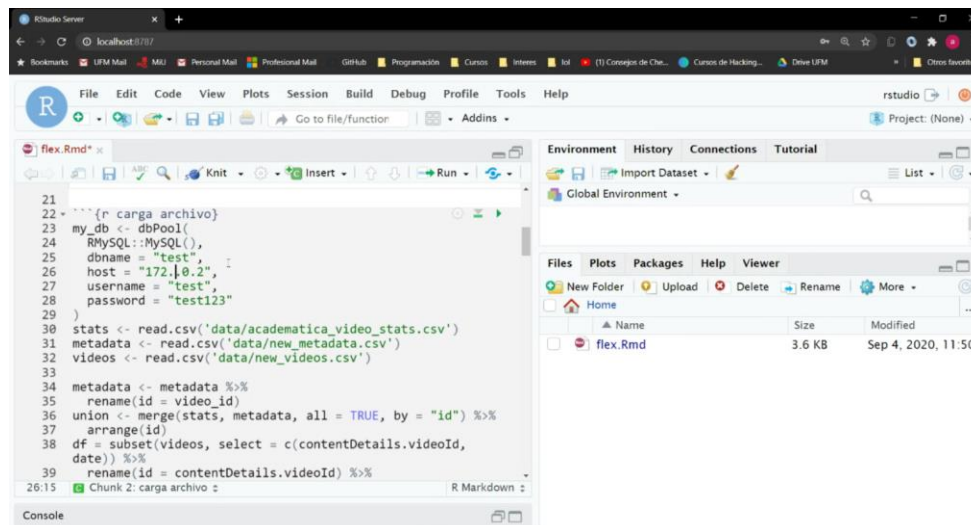






Ahora realizando la conexión para hacerlo desde Docker y la base de datos





Escribiendo a la base de datos

```
stats <- read.csv('data/academica_video_stats.csv')
metadata <- read.csv('data/new_metadata.csv')
videos <- read.csv('data/new_videos.csv')

videos <- dbWriteTable(basededatos, "x", archivocsv, append
= TRUE, row.names = FALSE)

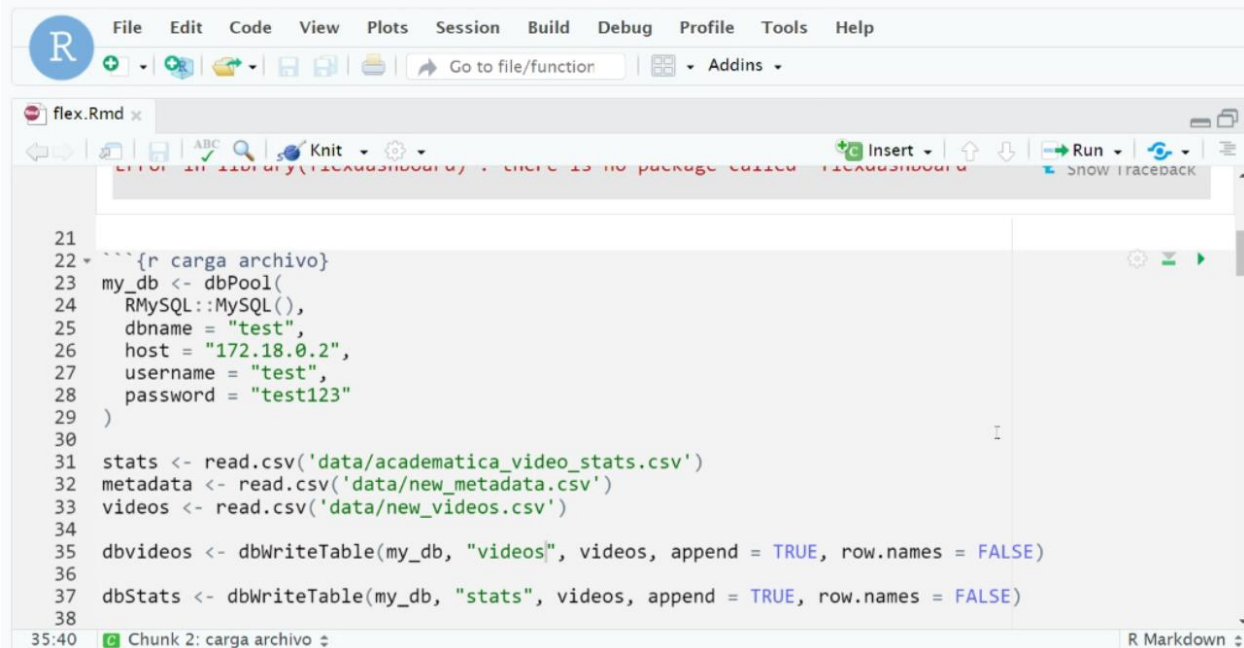
stats <- read.csv('data/academica_video_stats.csv')
metadata <- read.csv('data/new_metadata.csv')
videos <- read.csv('data/new_videos.csv')

dbvideos <- dbWriteTable(my_db, "stats", videos, append =
TRUE, row.names = FALSE)

dbStats <- dbWriteTable(my_db, "stats", videos, append =
TRUE, row.names = FALSE)

dbMeta <- dbWriteTable(my_db, "metadata", videos, append =
TRUE, row.names = FALSE)
```

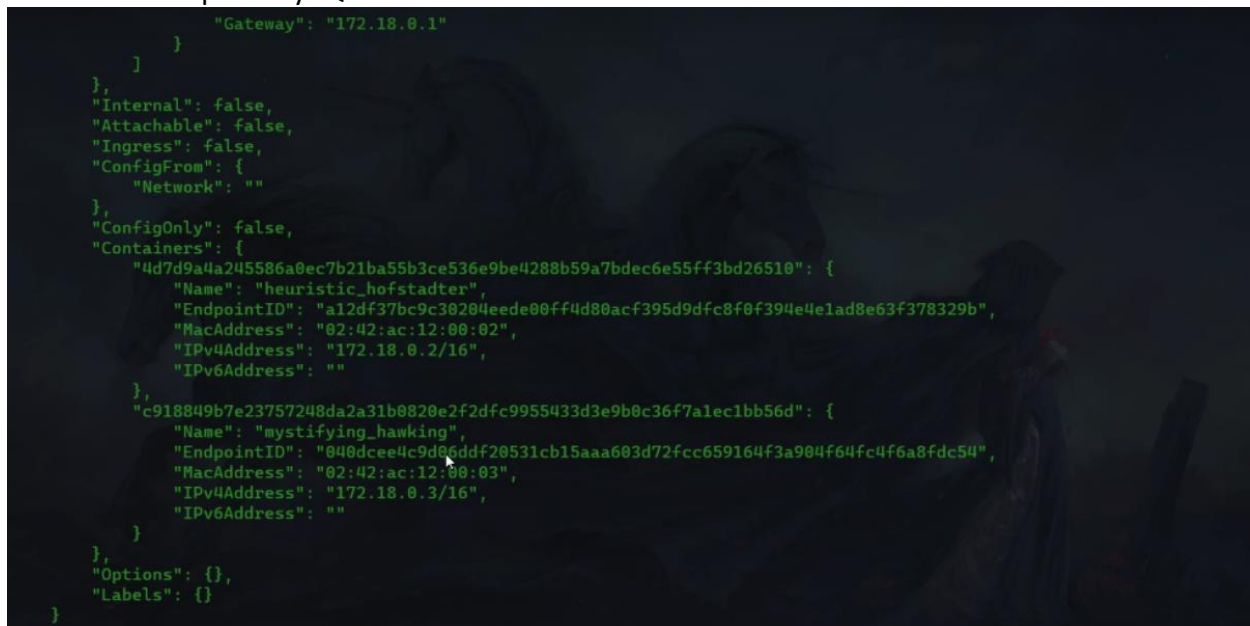
Populando las tablas



```
21
22 {r carga archivo}
23 my_db <- dbPool(
24 RMySQL::MySQL(),
25 dbname = "test",
26 host = "172.18.0.2",
27 username = "test",
28 password = "test123"
29)
30
31 stats <- read.csv('data/academica_video_stats.csv')
32 metadata <- read.csv('data/new_metadata.csv')
33 videos <- read.csv('data/new_videos.csv')
34
35 dbvideos <- dbWriteTable(my_db, "videos", videos, append = TRUE, row.names = FALSE)
36
37 dbStats <- dbWriteTable(my_db, "stats", videos, append = TRUE, row.names = FALSE)
38
```

35:40 Chunk 2: carga archivo R Markdown

Verificando el ip de MySQL



```
 "Gateway": "172.18.0.1"
 },
],
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {
 "Network": ""
 },
 "ConfigOnly": false,
 "Containers": {
 "4d7d9a4a245586a0ec7b21ba55b3ce536e9be4288b59a7bdec6e55ff3bd26510": {
 "Name": "heuristic_hofstadter",
 "EndpointID": "a12df37bc9c30204eede00ff4d80acf395d9dfc8f0f394e4e1ad8e63f378329b",
 "MacAddress": "02:42:ac:12:00:02",
 "IPv4Address": "172.18.0.2/16",
 "IPv6Address": ""
 },
 "c918849b7e23757248da2a31b0820e2f2dfc9955433d3e9b0c36f7a1ec1bb56d": {
 "Name": "mystifying_hawking",
 "EndpointID": "040dcee4c9d06ddf20531cb15aaa603d72fcc659164f3a904f64fc4f6a8fdc54",
 "MacAddress": "02:42:ac:12:00:03",
 "IPv4Address": "172.18.0.3/16",
 "IPv6Address": ""
 }
 },
 "Options": {},
 "Labels": {}
}
```

Para mas referencias pueden visitar el siguiente link:

<http://alexanderpalencia.me/Data-Product/#videos-vistos-1>