

# PROY-NB03-TWEETS

December 9, 2018

## 1 Proyecto final. Datamining.

### 1.0.1 Análisis tweets UEFA Champions League Final 2018

### 1.0.2 Participantes:

Gonzalo de las Heras de Matías - Jorge de la Fuente Tagarro - Alejandro Amarillas Cámara - Sergio Sampo Balmaseda.

### 1.0.3 Notebook (3/4). Análisis de sentimiento.

### 1.0.4 Objetivo del notebook:

Este notebook analiza el texto de los distintos tweets.

#### Librerías

```
In [1]: import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import gmaps
import gmaps.datasets
import gmaps.geojson_geometries
from collections import Counter
from Library.Apriori import APriori
from wordcloud import WordCloud, STOPWORDS

pd.options.display.max_colwidth = 500
gmaps.configure(api_key="AIzaSyChafUDb00RYZ7y7wsZKhgslvQXSDudURQ")
```

#### Carga de datos

```
In [2]: Datos = pd.read_json("datos_limpios.json")
print("Hay", len(Datos), "registros")
```

Hay 330384 registros

Desgranamos la hora de cada tweet para poder agrupar más fácilmente.



title

```
In [3]: Datos["hora"] = pd.to_datetime(Datos["hora"])
        Datos["min"] = Datos["hora"].dt.minute
        Datos["hour"] = Datos["hora"].dt.hour
        del Datos["hora"]
```

Eliminamos 61 registros que son antes del partido a una hora distanciada de la hora del partido.

```
In [4]: Datos = Datos[Datos["hour"] > 15]
        print("Hay", len(Datos), "registros")
```

Hay 330323 registros

## 2 1.- Mapa de palabras más usadas

Creamos un mapa de palabras con aquellas más usadas.

```
In [5]: palabras = []
        for line in Datos["tokens"] :
            palabras.extend(line)

        # Contamos las palabras.
        frecuenciaPalabras = Counter(palabras)

        # Creamos el mapa de palabras.
        wordcloud = WordCloud(width=1600,
                               height=800,
                               max_words=200,
```



```
In [6]: sentimientos = Datos.pivot_table(index=['hour', 'min'], columns='sentimiento_tipo', aggfunc='sum')
sentimientos = sentimientos.reset_index()
sentimientos.head()
```

```
Out [6]: sentimiento_tipo  hour  min  NEGATIVE  NEUTRAL  POSITIVE
0                20      45         12        170         74
1                20      46        196       1820        838
2                20      47        203       1685        839
3                20      48        290       1495        872
4                20      49        253       1430        817
```

Creamos una nueva tabla con toda la información anterior.

```
In [7]: sentimientos["hora_str"] = ""
sentimientos["hora"] = ""
for index, row in sentimientos.iterrows():
    if row["min"] < 10:
        minutos = "0" + str(row["min"])
    else:
        minutos = str(row["min"])
    sentimientos.loc[index, "hora_str"] = str(row["hour"]) + ":" + minutos
    sentimientos.loc[index, "hora"] = str(row["hour"]) + ":" + minutos + ":00"
sentimientos["hora_dt"] = pd.to_datetime(sentimientos["hora"], format="%H:%M:%S")
sentimientos.head()
```

```
Out [7]: sentimiento_tipo  hour  min  NEGATIVE  NEUTRAL  POSITIVE  hora_str  hora \
0                20      45         12        170         74      20:45  20:45:00
1                20      46        196       1820        838      20:46  20:46:00
2                20      47        203       1685        839      20:47  20:47:00
3                20      48        290       1495        872      20:48  20:48:00
4                20      49        253       1430        817      20:49  20:49:00

sentimiento_tipo  hora_dt
0                1900-01-01 20:45:00
1                1900-01-01 20:46:00
2                1900-01-01 20:47:00
3                1900-01-01 20:48:00
4                1900-01-01 20:49:00
```

### 3.1 2.1.- Primera parte

En la primera parte el evento más significativo es la lesión de Salah tras el choque con Sergio Ramos sobre el minuto 30 de partido.

Filtramos la tabla de datos.

```
In [8]: primeraParte = sentimientos[sentimientos["hora_dt"] < "1900-01-01 21:45:00"]
primeraParte.head()
```

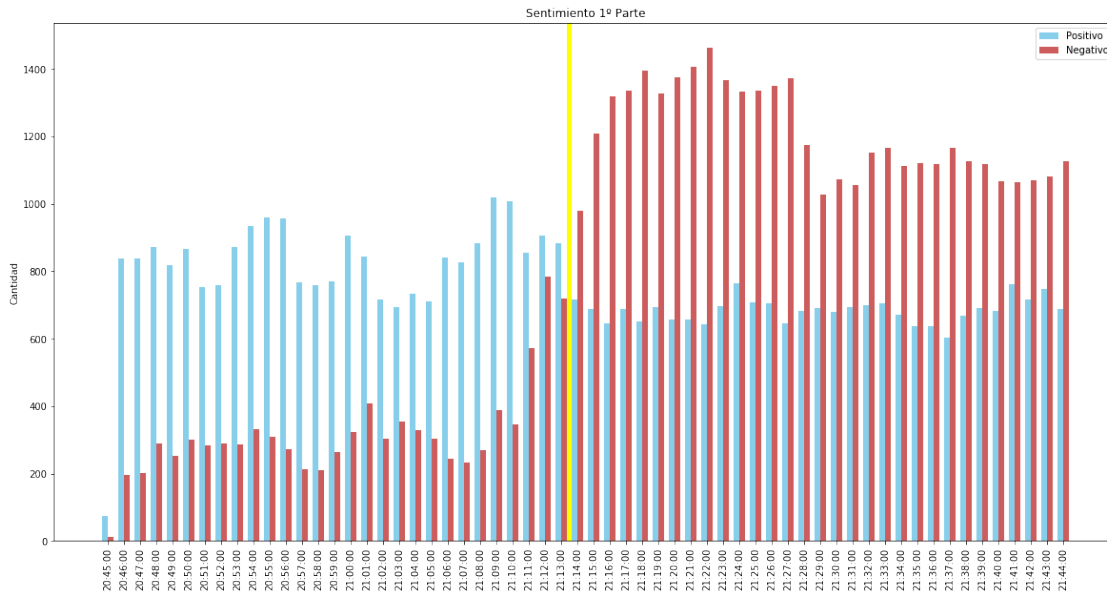
```
Out[8]: sentimiento_tipo  hour  min  NEGATIVE  NEUTRAL  POSITIVE  hora_str  hora  \
0                20    45         12        170         74    20:45  20:45:00
1                20    46        196       1820        838    20:46  20:46:00
2                20    47        203       1685        839    20:47  20:47:00
3                20    48        290       1495        872    20:48  20:48:00
4                20    49        253       1430        817    20:49  20:49:00

sentimiento_tipo          hora_dt
0                1900-01-01 20:45:00
1                1900-01-01 20:46:00
2                1900-01-01 20:47:00
3                1900-01-01 20:48:00
4                1900-01-01 20:49:00
```

Creamos una tabla para observar la variación del sentimiento durante la primera parte.

```
In [9]: # Preparamos la gráfica.
ind = np.arange(len(primeraparte))
ancho = 0.35
fig, ax = plt.subplots(figsize=(20, 10))
rects1 = ax.bar(ind - ancho/2, primeraParte["POSITIVE"], ancho, color='SkyBlue', label='POSITIVE')
rects2 = ax.bar(ind + ancho/2, primeraParte["NEGATIVE"], ancho, color='IndianRed', label='NEGATIVE')
ax.set_ylabel('Cantidad')
ax.set_title('Sentimiento 1ª Parte')
ax.set_xticks(ind)
ax.set_xticklabels(primeraparte["hora"])
plt.axvline(x=28.5, color="yellow", linewidth=5)
ax.legend()
plt.xticks(rotation=90)

# Mostramos la gráfica.
plt.show()
```



La línea amarilla indica el momento en el que se lesionó Salah.

Observamos como después de la lesión de Salah, los comentarios negativos se disparan. Imprimimos algunos de estos comentarios.

```
In [10]: comentarios_neg = pd.DataFrame(Datos[(Datos["hour"] == 21) & (Datos["min"] == 12) & \
                                             (Datos["sentimiento_tipo"] == "NEGATIVE")]["texto"]
                                             .reset_index(drop=True)
comentarios_neg[(comentarios_neg["texto_traducido"].str.contains("ramos")) |
                 (comentarios_neg["texto_traducido"].str.contains("sergio"))].head()
```

Out [10]:

```
1
2
14      sergio ramos applies a wrestling key mohamed salah egyptian ends affected c
15 sergio ramos just dragged salah the ground the arm now salah down clutching his sl
16                                     ramos the worst
```

Los comentarios negativos hacia Ramos continúan mucho después. Los aficionados se mostraron muy enfadados con el suceso.

```
In [11]: comentarios_neg = pd.DataFrame(Datos[(Datos["hour"] == 21) & (Datos["min"] == 44) & \
                                             (Datos["sentimiento_tipo"] == "NEGATIVE")]["texto"]
                                             .reset_index(drop=True)
comentarios_neg[(comentarios_neg["texto_traducido"].str.contains("ramos")) |
                 (comentarios_neg["texto_traducido"].str.contains("sergio"))].head()
```

Out [11]:

```
0                                     ronaldo knew what ramos wa
1 breaking news this moment world most hated person not donald
```

```

2 sergio ramos fraud sergio ramos should have been wrestler purposely injured salah
4
6
retweet you think sergio ramos inj

```

## 3.2 2.2.- Segunda parte

En la segunda parte los eventos más significativos son:

```

<li>21:50. Min. 50. Gol de Benzema. Error de Karius.</li>
<li>21:54. Min. 54. Gol de Mané.</li>
<li>22:03. Min. 63. Gol de Bale. Chilena</li>
<li>22:09. Min. 69. Palo del Liverpool.</li>
<li>22:23. Min. 83. Gol de Bale. Error de Karius.</li>

```

Filtramos la tabla de datos.

```

In [12]: segundaParte = sentimientos[sentimientos["hora_dt"] >= "1900-01-01 21:45:00"]
        segundaParte.head()

Out[12]: sentimiento_tipo  hour  min  NEGATIVE  NEUTRAL  POSITIVE  hora_str  hora  \
60                        21    45      1202      991      685    21:45  21:45:00
61                        21    46      1118     1043      712    21:46  21:46:00
62                        21    47      1106     1007      773    21:47  21:47:00
63                        21    48      1079     1027      776    21:48  21:48:00
64                        21    49      1083     1012      789    21:49  21:49:00

        sentimiento_tipo  hora_dt
60                    1900-01-01 21:45:00
61                    1900-01-01 21:46:00
62                    1900-01-01 21:47:00
63                    1900-01-01 21:48:00
64                    1900-01-01 21:49:00

```

Creamos una tabla para observar la variación del sentimiento durante la segunda parte.

```

In [13]: # Preparamos la gráfica.
        ind = np.arange(len(segundaParte))
        ancho = 0.35
        fig, ax = plt.subplots(figsize=(20, 10))
        rects1 = ax.bar(ind - ancho/2, segundaParte["POSITIVE"], ancho, color='SkyBlue', label='POSITIVE')
        rects2 = ax.bar(ind + ancho/2, segundaParte["NEGATIVE"], ancho, color='IndianRed', label='NEGATIVE')
        ax.set_xticklabels(primeraparte["hora"])
        plt.axvline(x=4.5, color="yellow",linewidth=5)
        plt.axvline(x=8.5, color="yellow",linewidth=5)
        plt.axvline(x=17.5, color="yellow",linewidth=5)
        plt.axvline(x=23.5, color="yellow",linewidth=5)
        plt.axvline(x=37.5, color="yellow",linewidth=5)
        plt.axvline(x=44.5, color="yellow",linewidth=5)
        ax.set_ylabel('Cantidad')

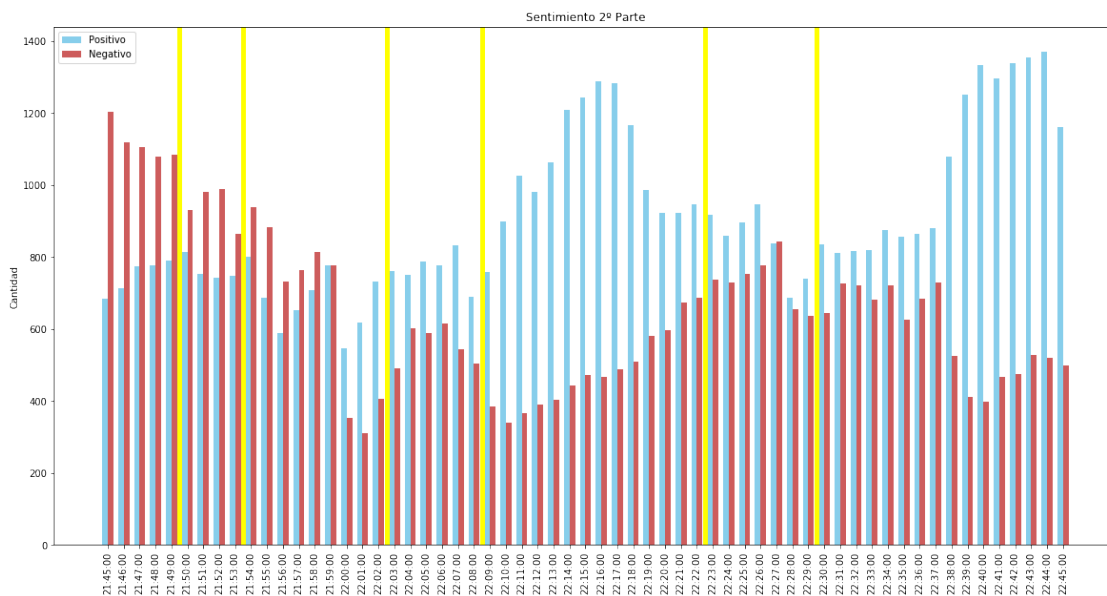
```

```

ax.set_title('Sentimiento 2ª Parte')
ax.set_xticks(ind)
ax.set_xticklabels(segundaParte["hora"])
ax.legend()
plt.xticks(rotation=90)

# Mostramos la gráfica.
plt.show()

```



La líneas amarillas indican:

```

<li>Gol de Benzema. Error de Karius.</li>
<li>Gol de Mané.</li>
<li>Gol de Bale. Chilena</li>
<li>Palo del Liverpool.</li>
<li>Gol de Bale. Error de Karius.</li>
<li>Fin del partido.</li>

```

A continuación, analizamos el comportamiento de los tweeteros con respecto a cada evento.

### 3.2.1 2.2.1.- Min. 50. 21:50. Gol de Benzema. Error de Karius.

```

In [14]: comentarios_neg = pd.DataFrame(Datos[(Datos["hour"] == 21) & (Datos["min"] <= 59) & \
                                             (Datos["min"] > 50) & \
                                             (Datos["sentimiento_tipo"] == "NEGATIVE")]["texto"]
                                             .reset_index(drop=True)
comentarios_neg[comentarios_neg["texto_traducido"].str.contains("karius")].head()

```

Out[14]:

```

5          between ulrich karius will turn a season shit legendar

```



```

8                                                                    karius fre
9                                goooaaal real madrid liverpool benzema blocks karius throw and ro
14
24  direct gooooooooooooooooool benzema karius takes out with a bad hand golepa foot of tl

```

Los tweets van en contra de Karius.

### 3.2.2 2.2.2.- Min. 54. 21:54. Gol de Mané.

```

In [15]: comentarios_neg = pd.DataFrame(Datos[(Datos["hour"] == 21) & (Datos["min"] <= 59) & \
                                             (Datos["sentimiento_tipo"] == "NEGATIVE")]["texto"]
                                             .reset_index(drop=True)
comentarios_neg[(comentarios_neg["texto_traducido"].str.contains("sadio")) |
                 (comentarios_neg["texto_traducido"].str.contains("mane"))].head()

```

Out [15]:

```

2356                                proper snide wanker that sergio ramos lil
3079                                wrong cry and try to be persuaded mane seems wrong journey
4231  real will be leading goal mane in the beginning game sallah will be out on injur
4330                                ramos should banned
4386  real will be leading goal mane in the beginning game sallah will be out on injur

```

Predominan los comentarios negativos. En los tweets la gente se acuerda del incidente entre Ramos y Salah.

### 3.2.3 2.2.3.- Min. 60. 22:00. Gol de Bale. Chilena

```

In [16]: comentarios_neg = pd.DataFrame(Datos[(Datos["hour"] == 22) & (Datos["min"] == 10) & \
                                             (Datos["sentimiento_tipo"] == "NEGATIVE")]["texto"]
                                             .reset_index(drop=True)
comentarios_neg[(comentarios_neg["texto_traducido"].str.contains("bale")) |
                 (comentarios_neg["texto_traducido"].str.contains("karius"))].head()

```

Out [16]:

```

                                texto_traducido
3                                omfg wtf was tht bale\n
10  hate say but damn bale that was incredible
11                                that ridiculous from gareth bale
13                                holy shit bale why why
20                                chuffin hell bale

```

Predominan los comentarios negativos. Los tweets se clasifican como negativos porque claman contra el gol de Bale. Implícitamente reconocen que es un gran gol.

### 3.2.4 2.2.4.- Min. 83. 22:23. Gol de Bale. Error de Karius.

```

In [17]: comentarios_neg = pd.DataFrame(Datos[(Datos["hour"] == 22) & (Datos["min"] == 30) & \
                                             (Datos["sentimiento_tipo"] == "NEGATIVE")]["texto"]
                                             .reset_index(drop=True)
comentarios_neg[(comentarios_neg["texto_traducido"].str.contains("bale")) |
                 (comentarios_neg["texto_traducido"].str.contains("karius"))].head(10)

```

```

Out[17]:
0
1
8
9
10
12
15
28
29
32
    texto_tradu
does there need match fixing investigation karius just that sl
    karius shit part lmaoooo
karius just emphatically ended his career disap
I just feel sorry for k
where fuck did they find that kar
all human and all make mistakes but karius has made enough for all toni
karius career is likely to be finished first poor
sure karius wants crawl into hole and d
karius with night forget but hats off gareth bale for coming off the bench score t
somebody tell karius not playing sunday league football\nbloo

```

Predominan los comentarios negativos. Una vez más, los tweets van en contra de Karius.

## 4 3.- Comparativa Ramos/Salah, Bale y Karius

Obtenemos los datos necesarios

```

In [18]: # Sergio ramos
ramos = pd.DataFrame(Datos[(Datos["texto_traducido"].str.contains("sergio")) |
    (Datos["texto_traducido"].str.contains("ramos")) |
    (Datos["texto_traducido"].str.contains("salah"))]) \
    .reset_index(drop=True)
ramos[['hour', 'min', 'sentimiento_tipo']]
ramos = ramos.pivot_table(index=['hour', 'min'], columns='sentimiento_tipo', aggfunc='s
ramos = ramos.reset_index()
ramos["hora_str"] = ""
ramos["hora"] = ""
for index, row in ramos.iterrows():
    if row["min"] < 10:
        minutos = "0" + str(row["min"])
    else:
        minutos = str(row["min"])
    ramos.loc[index, "hora_str"] = str(row["hour"]) + ":" + minutos
    ramos.loc[index, "hora"] = str(row["hour"]) + ":" + minutos + ":00"
ramos["hora_dt"] = pd.to_datetime(ramos["hora"], format="%H:%M:%S")

# Bale
bale = pd.DataFrame(Datos[(Datos["texto_traducido"].str.contains("bale")) |
    (Datos["texto_traducido"].str.contains("gareth"))]) \
    .reset_index(drop=True)
bale[['hour', 'min', 'sentimiento_tipo']]
bale = bale.pivot_table(index=['hour', 'min'], columns='sentimiento_tipo', aggfunc='s
bale = bale.reset_index()
bale["hora_str"] = ""
bale["hora"] = ""
for index, row in bale.iterrows():
    if row["min"] < 10:

```

```

        minutos = "0" + str(row["min"])
    else:
        minutos = str(row["min"])
    bale.loc[index, "hora_str"] = str(row["hour"]) + ":" + minutos
    bale.loc[index, "hora"] = str(row["hour"]) + ":" + minutos + ":00"
bale["hora_dt"] = pd.to_datetime(bale["hora"], format="%H:%M:%S")

# Karius
karius = pd.DataFrame(Datos[(Datos["texto_traducido"].str.contains("karius")) |
                             (Datos["texto_traducido"].str.contains("loris"))]) \
        .reset_index(drop=True)
karius[["hour", 'min', 'sentimiento_tipo']]
karius = karius.pivot_table(index=['hour', 'min'], columns='sentimiento_tipo', aggfun
karius = karius.reset_index()
karius["hora_str"] = ""
karius["hora"] = ""
for index, row in karius.iterrows():
    if row["min"] < 10:
        minutos = "0" + str(row["min"])
    else:
        minutos = str(row["min"])
    karius.loc[index, "hora_str"] = str(row["hour"]) + ":" + minutos
    karius.loc[index, "hora"] = str(row["hour"]) + ":" + minutos + ":00"
karius["hora_dt"] = pd.to_datetime(karius["hora"], format="%H:%M:%S")

```

#### 4.1 3.1.- Sergio Ramos/Salah

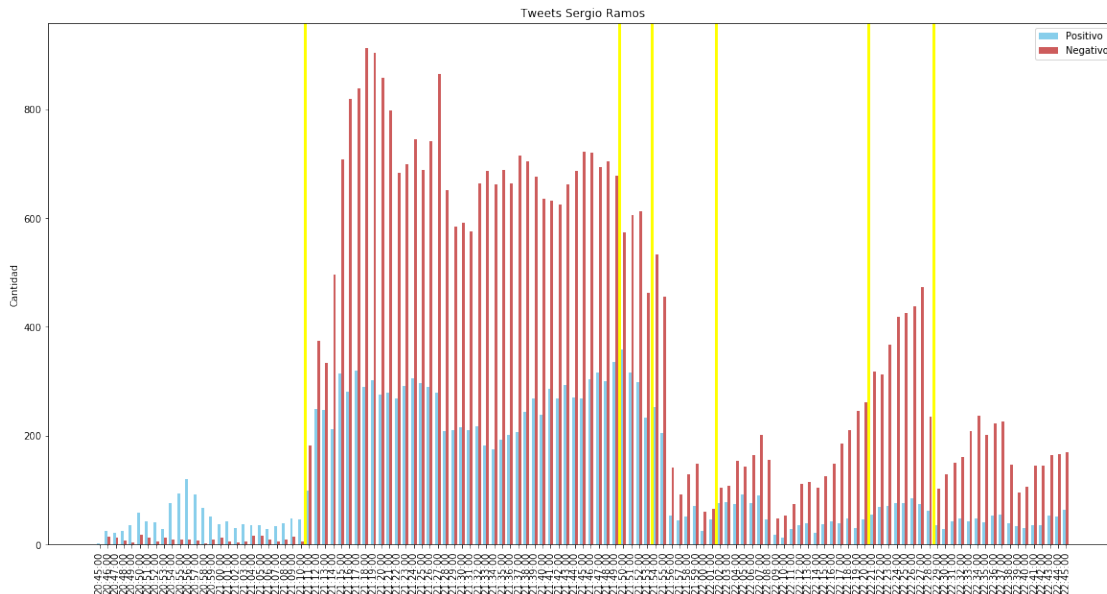
```

In [19]: # Preparamos la gráfica.
ind = np.arange(len(ramos))
ancho = 0.35
fig, ax = plt.subplots(figsize=(20, 10))
rects1 = ax.bar(ind - ancho/2, ramos["POSITIVE"], ancho, color='SkyBlue', label='Posi
rects2 = ax.bar(ind + ancho/2, ramos["NEGATIVE"], ancho, color='IndianRed', label='Neg
ax.set_xticklabels(primeraparte["hora"])
plt.axvline(x=25.5, color="yellow",linewidth=3)
plt.axvline(x=64.5, color="yellow",linewidth=3)
plt.axvline(x=68.5, color="yellow",linewidth=3)
plt.axvline(x=76.5, color="yellow",linewidth=3)
plt.axvline(x=95.5, color="yellow",linewidth=3)
plt.axvline(x=103.5, color="yellow",linewidth=3)
ax.set_ylabel('Cantidad')
ax.set_title('Tweets Sergio Ramos')
ax.set_xticks(ind)
ax.set_xticklabels(ramos["hora"])
ax.legend()
plt.xticks(rotation=90)

# Mostramos la gráfica.

```

```
plt.show()
```



La líneas amarillas indican:

```
<li>Lesión de Salah.</li>
<li>Gol de Benzema. Error de Karius.</li>
<li>Gol de Mané.</li>
<li>Gol de Bale. Chilena</li>
<li>Gol de Bale. Error de Karius.</li>
<li>Fin del partido.</li>
```

Se habla de Ramos en el momento que se lesiona Salah hasta que marca Mané. Luego hay un repunte con el error de Karius en el tercer gol al final del partido.

## 4.2 3.2.- Karius

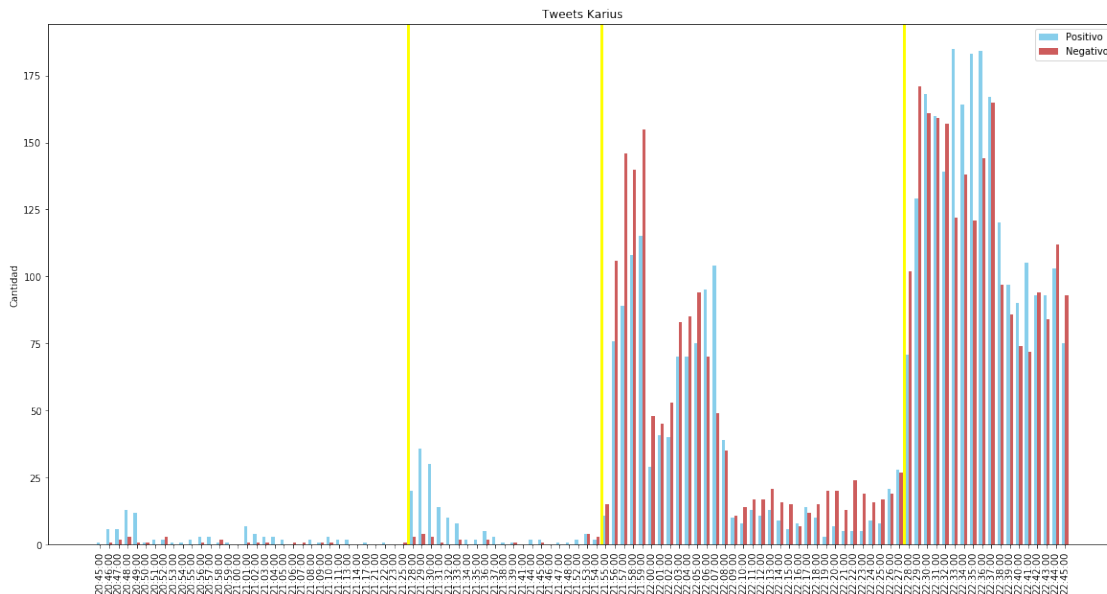
```
In [20]: # Preparamos la gráfica.
ind = np.arange(len(karius))
ancho = 0.35
fig, ax = plt.subplots(figsize=(20, 10))
rects1 = ax.bar(ind - ancho/2, karius["POSITIVE"], ancho, color='SkyBlue', label='Pos')
rects2 = ax.bar(ind + ancho/2, karius["NEGATIVE"], ancho, color='IndianRed', label='Neg')
ax.set_xticklabels(primeraparte["hora"])
plt.axvline(x=33.5, color="yellow",linewidth=3)
plt.axvline(x=54.5, color="yellow",linewidth=3)
plt.axvline(x=87.5, color="yellow",linewidth=3)
ax.set_ylabel('Cantidad')
ax.set_title('Tweets Karius')
ax.set_xticks(ind)
```

```

ax.set_xticklabels(karius["hora"])
ax.legend()
plt.xticks(rotation=90)

# Mostramos la gráfica.
plt.show()

```



La primera línea amarilla indican una parada de mucho mérito a CR7 y el resto indican los instantes en los que Karius cometió sus errores.

Se habla de Karius positivamente cuando realizó una buena parada a un remate de Cristiano Ronaldo, pero en sus 2 errores, se habla mucho y negativamente.

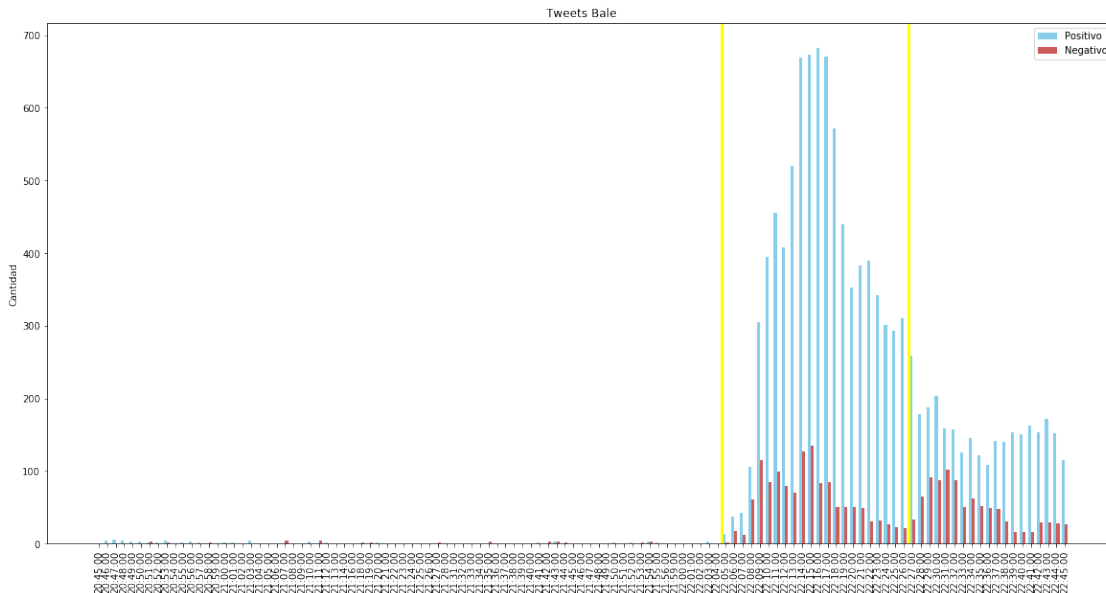
### 4.3 3.3.- Bale

```

In [21]: # Preparamos la gráfica.
ind = np.arange(len(bale))
ancho = 0.35
fig, ax = plt.subplots(figsize=(20, 10))
rects1 = ax.bar(ind - ancho/2, bale["POSITIVE"], ancho, color='SkyBlue', label='Posit')
rects2 = ax.bar(ind + ancho/2, bale["NEGATIVE"], ancho, color='IndianRed', label='Neg')
ax.set_xticklabels(primeraparte["hora"])
plt.axvline(x=73.5, color="yellow",linewidth=3)
plt.axvline(x=95.5, color="yellow",linewidth=3)
ax.set_ylabel('Cantidad')
ax.set_title('Tweets Bale')
ax.set_xticks(ind)
ax.set_xticklabels(bale["hora"])
ax.legend()
plt.xticks(rotation=90)

```

```
# Mostramos la gráfica.
plt.show()
```



La líneas amarillas indican los goles de Bale.

Se habla mucho de Bale desde que sale al campo y mete el primer gol. El segundo no se habla tanto ya que se achaca más a Karius.

## 5 4.- Tweets Cataluña

```
In [22]: # Capturamos los datos necesarios.
```

```
catalunya = pd.DataFrame(Datos[Datos["idioma"] == "ca"]) \
    .reset_index(drop=True)
catalunya[["hour", "min", "sentimiento_tipo"]]
catalunya = catalunya.pivot_table(index=["hour", "min"], columns="sentimiento_tipo", )
catalunya = catalunya.reset_index()
catalunya["hora_str"] = ""
catalunya["hora"] = ""
for index, row in catalunya.iterrows():
    if row["min"] < 10:
        minutos = "0" + str(row["min"])
    else:
        minutos = str(row["min"])
    catalunya.loc[index, "hora_str"] = str(row["hour"]) + ":" + minutos
    catalunya.loc[index, "hora"] = str(row["hour"]) + ":" + minutos + ":00"
catalunya["hora_dt"] = pd.to_datetime(ramos["hora"], format="%H:%M:%S")

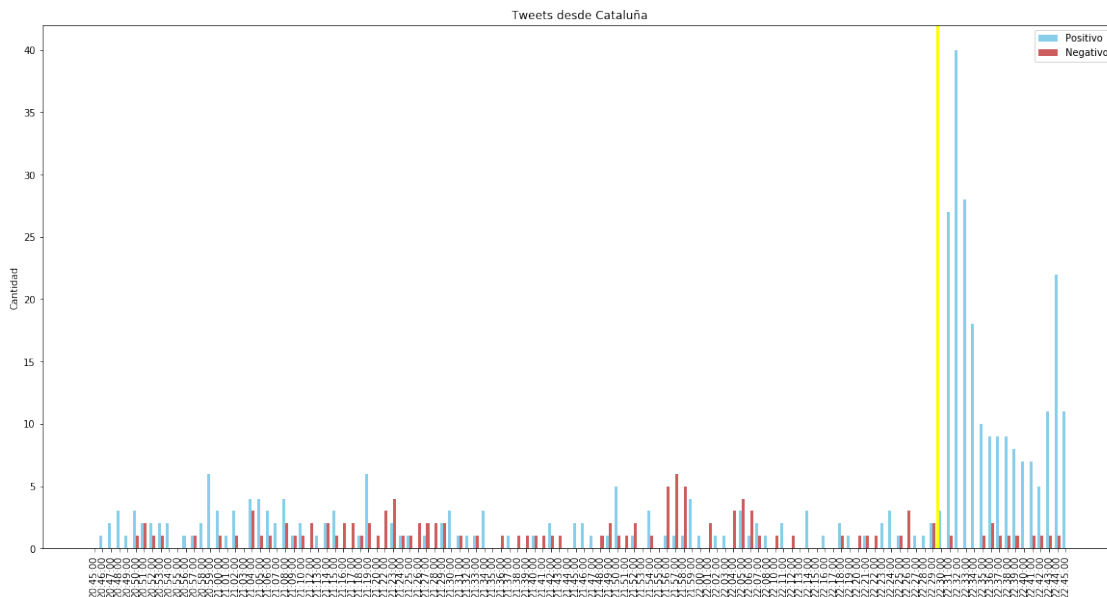
# Preparamos la gráfica.
```

```

ind = np.arange(len(catalunya))
ancho = 0.35
fig, ax = plt.subplots(figsize=(20, 10))
rects1 = ax.bar(ind - ancho/2, catalunya["POSITIVE"], ancho, color='SkyBlue', label='Positivo')
rects2 = ax.bar(ind + ancho/2, catalunya["NEGATIVE"], ancho, color='IndianRed', label='Negativo')
ax.set_xticklabels(catalunya["hora"])
ax.set_ylabel('Cantidad')
ax.set_title('Tweets desde Cataluña')
ax.set_xticks(ind)
ax.set_xticklabels(catalunya["hora"])
plt.axvline(x=101.5, color="yellow",linewidth=3)
ax.legend()
plt.xticks(rotation=90)

# Mostramos la gráfica.
plt.show()

```



La línea amarilla indica el final del partido.

Resulta llamativo que el eterno rival del Real Madrid no escriba demasiados tweets en contra de ellos. De hecho, al final de partido, los aficionados del Madrid catalanes salen a la luz con comentarios y retweets positivos. Vemos unos ejemplos.

```

In [23]: comentarios = pd.DataFrame(Datos[(Datos["hour"] == 22) & (Datos["min"] >= 34) \
                                         & (Datos["idioma"] == "ca")]["texto_traducido"])
                                         .reset_index(drop=True)

comentarios.head(20)

```

Out [23]:

```

0 tridente del madrid this champion

```

1	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
2	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
3	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
4	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
5	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
6	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
7									enter	emre	can to get used t
8									tridente del	madrid	this champions
9	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
10									after bribing the average arbitrators uefa press this year have bribed		
11											
12	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
13	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
14											
15	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
16									I&#39;m going to found		
17	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
18	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018
19	final	champions	2014	goal	final	champions	2016	attendance	final	champions	2018

## 6 Referencias

<https://www.kaggle.com/xvivancos/tweets-during-r-madrid-vs-liverpool-ucl-2018>  
<https://github.com/pbugnion/gmaps>  
<https://pandas.pydata.org/>  
 Apuntes de la asignatura