



Examen Simulacion

Docente:

Ing. Diego Quisi.

Alumno:

Juan Carlos Cañar Uyaguari.

21/12/2020 .

Introducción:

El golpe económico de la crisis sanitaria del corona virus no va a ser cosa de semanas, sino de meses. Dentro de una de las etapas importantes que están a la vuelta de la esquina son las elecciones presidenciales y assembleístas del Ecuador. Para ello se plantea realizar un sistema de regresión que permita identificar cual es la tendencia de los votos en base al manejo de las redes sociales (Twitter y/o Facebook) [1].

Las regresiones lineales pueden aprender por sí mismos y en este caso obtener automáticamente esa “recta” que buscamos con la tendencia de predicción. Para hacerlo se mide el error con respecto a los puntos de entrada y el valor “Y” de salida real [3].

Enunciado:

- Diseñe y desarrolle un modelo y/o script que permita simular el siguiente caso real: ◦ Obtener datos de tendencia de twitter o facebook, para ello se puede obtener a través del API [4].
- Title: Título del Post/Twitter
- Word count: la cantidad de palabras del artículo,
- # of Links: los enlaces externos que contiene,
- # of comments: cantidad de comentarios,
- # Shares: compartidos.
- HashTag • Etc.

En base a ello, se pretende proponer y generar una predicción de cuántas veces será compartido un post/twitter utilizando regresión [2].

- Posteriormente se debe seguir un procesos de votación de eventos discretos que se describe a continuación:
 - Tomar los resultados de la regresión para la selección del candidato.
 - Se tiene una tendencia del 90% de personas que realizan el proceso de elección dentro del Ecuador.
 - Dentro del procesos se tiene que alrededor del 5% - 10% votan nulo.
 - Solo se va a tener en cuenta las elecciones de los asambleístas por el Azuay.
 - Las personas solo tiene un recinto electoral para realizar el proceso.
 - Las personas solo pueden realizar un proceso de elección por asambleísta del Azuay.
 - La persona se acerca a la mesa electoral y hacen fila en caso de ser necesario.
 - Realiza el voto en un tiempo aleatorio de un partido específico.
 - La persona recibe su certificado votación.
- La persona sale del recinto electoral.
- Finalmente generar una grafica de las personas que votaron y los asambleístas electos.
- El proceso de simulación desarrollado deberá considerar los siguientes aspectos:
 - Se debe establecer un modelo basado en modelos matemáticos y probabilísticos para la predicción del numero de veces que se compartirá o la tendencia electoral basada en redes sociales.
 - El programa deberá generar gráficas que indiquen la ecuación matemática y probabilística de tendencias de votaciones.
 - Deben calcularse las siguientes métricas del sistema de simulación de eventos discretos :
 - Total de de personas que realizaron el proceso de votación.
 - Asambleístas ganadores.
 - El tiempo promedio de espera.
- Fecha de presentación: 20/12/2020 – 23:55.
- El informe en pdf (Archivo Notebook) se deberá subir al Git personal, todo el sistema debe ser basado en Python - Open Source.
- Cualquier adición o librerías que no se revisaron en clases serán tomada como puntos adicionales al interciclo.

- **API key**

- 4JnLpCHZrSNWA9udcVjdoXPYf

- **API secret key**

- 3sYWK7vpVCkls36iyf8EVGcPxFmT6F2pXTEZX5PhHxwpyTbEgg

- **Bearer token**

- AAAAAAAAAAAAAAAAAAAAAAAC0mKwEAAAAAYgkTE3LFoSVRw8ZahWZH4v5eNv0%3DuXuKi6XaoL2

In [162]:

```
import numpy as np
from datetime import datetime, timedelta
from sklearn.metrics import mean_squared_error
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
from sklearn import linear_model
%matplotlib inline
from xml.dom import minidom
import json
import tweepy as tw
import pandas as pd
import os
import sys
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from datetime import datetime, timedelta
```

In [32]:

```
consumer_key = "4JnLpCHZrSNWA9udcVjdoXPYf"
consumer_secret = "3sYWK7vpVCKls36iyf8EVGcPxFmT6F2pXTEZX5PhHxwpyTbEgg"
access_token = "202024170-280kigS0W8n2icIacC0oapfF0Y70LJU86wdQlFtw"
access_token_secret = "CznuxrXC3zCKSGN9e1TBhqRko7geQY3BufGmY2RgQe5qr"

auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tw.API(auth)
```

In [2]:

```
results = api.trends_place(id = 375732, exclude = "hashtags")

trends = json.loads(json.dumps(results, indent=1))

for trend in trends[0]["trends"]:

    data = pd.DataFrame({'URL':trend['url'],'Nombre':trend['name'],'':trend['tweet_volume']})
display(data)
```

	URL	Nombre
1	http://twitter.com/search?q=Rafael	Rafael 120877

In [3]:

```
search_words = "Ecuador"
date_since = "2020-05-14"
new_search = search_words + " -filter:politica"
tweets = api.search(
    q=new_search,
    lang="es",
    result_type='popular')

#tweets
users_locs = [[tweet.user.screen_name, tweet.user.location,tweet.user.favourites_count,tweet.user.location] for tweet in tweets]
users_locs
```

Out[3]:

```
[['petrogustavo', 'ÜT: 4.650541,-74.074043', 16380, 130454, 4020319],
 ['Lenin', 'Ecuador', 162, 4173, 1066954],
 ['ecuarauz', 'Ecuador', 4026, 14615, 55883],
 ['MashiRafael', 'Ecuador', 570, 63165, 3740584],
 ['MashiRafael', 'Ecuador', 570, 63165, 3740584],
 ['evoespueblo', '', 139, 16169, 1182148],
 ['CELAGeopolitica', 'América Latina', 3597, 19441, 37614],
 ['AlvaroUribeVel', 'Colombia', 238, 85269, 4954083],
 ['elcomerciocom', 'Ecuador', 1376, 276157, 1787836],
 ['elcomerciocom', 'Ecuador', 1376, 276157, 1787836],
 ['SalvadorHeresi', 'Lima, Perú', 6814, 21887, 59538],
 ['VillaFernando_', 'Ecuador', 34826, 27994, 56447],
 ['Expresoec', 'Guayaquil', 1422, 171700, 786365],
 ['CarlosVerareal', 'Guayaquil', 737, 250305, 572402],
 ['LassoMusica', 'Mexico', 5047, 36439, 1253876]]
```

In [34]:

```
df = pd.DataFrame({'Nombre': list(map(lambda x: x.author.name , tweets)),
                  'Tweet': list(map(lambda x: x.text, tweets)),
                  'rt': list(map(lambda x: x.retweet_count, tweets)),
                  'favoritos': list(map(lambda x: x.favorite_count, tweets)),
                  'seguidores': list(map(lambda x: x.author.followers_count, tweets)),
                  'Caracteres': list(map(lambda x: len(x.text), tweets)),
                  'fecha': list(map(lambda x: x.created_at, tweets))})
df['fecha'] =pd.to_datetime(df.fecha)
df
```

Out[34]:

	Nombre	Tweet	rt	favoritos	seguidores	Caracteres	fecha
0	Gustavo Petro	El progresista Andrés Arauz ganaria la preside...	681	1607	4020319	140	2020-12-19 22:12:19
1	Lenín Moreno	¡Nuestro Ecuador se hace sentir! Gracias Carlo...	1821	1786	1066954	140	2020-12-19 15:01:18
2	Andrés Arauz	Tengo la responsabilidad de comunicar al Ecuad...	1119	3237	55883	139	2020-12-19 16:32:20
3	Rafael Correa	Ola Bini:\nTe pido disculpas en nombre de ECUA...	748	1769	3740584	140	2020-12-19 22:26:54
4	Rafael Correa	👤👤 Carlos Góngora se consagra como el primer ca...	425	1591	3740584	140	2020-12-19 06:43:44
5	Evo Morales Ayma	En San Benito se creó una comisión integrada p...	366	507	1182148	140	2020-12-20 13:36:34
6	CELAG	ec Andrés Arauz encabeza las preferencias el...	488	982	37614	140	2020-12-19 22:08:00
7	Álvaro Uribe Vélez	Al fondo Tulcán, Ecuador, visto desde Nariño h...	173	1271	4954083	68	2020-12-20 11:14:13
8	El Comercio	"Profe Nancy, buenas tardes. No puedo mandar e...	133	468	1787836	139	2020-12-19 17:36:57
9	El Comercio	Ministerio de Educación destacó la responsabil...	161	739	1787836	139	2020-12-19 22:00:00
10	Salvador Heresi	Mientras Chile, Colombia y Ecuador ya compraro...	123	401	59538	140	2020-12-19 13:19:16
11	Fernando Villavicencio	Maradona hizo con el chavismo lo mismo que el ...	94	178	56447	140	2020-12-19 16:34:16
12	Diario Expreso	El candidato a la presidencia de Ecuador, por ...	108	132	786365	140	2020-12-19 17:38:52

	Nombre	Tweet	rt	favoritos	seguidores	Caracteres	fecha
13	Carlos Vera	Ser un mejor Ecuador o terminar peor con los q...	144	830	572402	116	2020-12-19 13:19:23
14	Lassito	Horarios de Un Cuento de Navidad:\n\n6pm mx Méx...	72	817	1253876	140	2020-12-19 15:25:09

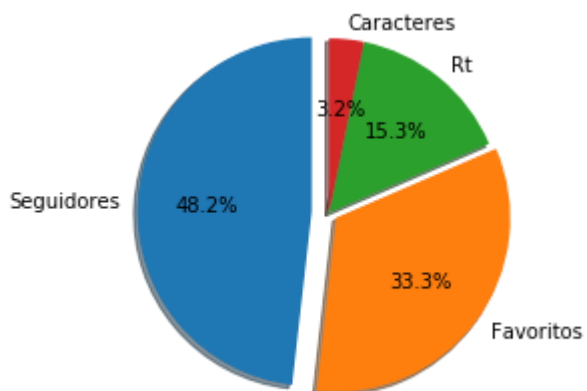
In [35]:

```
df1 = df.iloc[:, 2:6].cumsum(skipna=False)
#df1.plot(x='seguidores', y=['favoritos', 'rt', 'Caracteres'], figsize=(5,5))
s = df1['seguidores'].sum()
f = df1['favoritos'].sum()
r = df1['rt'].sum()
c = df1['Caracteres'].sum()
print(s,f,r,c)
Data = {'Tasks': [s,f,r,c]}
df = DataFrame(Data)

# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = 'Seguidores', 'Favoritos', 'Rt', 'Caracteres'
sizes = [s/1000, f, r, c]
explode = (0.1, 0.05, 0, 0) # only "explode" the 2nd slice (i.e. 'Hogs')

fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, explode=explode, autopct='%1.1f%%', shadow=True, startangle=90)
plt.show()
```

242317843 167295 76915 16150



In [54]:



```
#Sumamos el numero de seguidores + numero de rt y favoritos para obtener un solo punto  
df1["tot"] = df1[['favoritos', 'rt', 'Caracteres']].sum(axis=1)  
df2 = df1.iloc[:, [2,4]]  
df2
```

Out[54]:

	seguidores	sum
0	4020319	2428
1	5087273	6175
2	5143156	10670
3	8883740	13327
4	12624324	15483
5	13806472	16496
6	13844086	18106
7	18798169	19618
8	20586005	20358
9	22373841	21397
10	22433379	22061
11	22489826	22473
12	23276191	22853
13	23848593	23943
14	25102469	24972

- Regresion lineal

In [70]:

```
x = list(df2.iloc[:, 0]) # Fecha
y = list(df2.iloc[:, 1]) # Numero de casos
# Creamos el objeto de Regresión Lineal
regr = linear_model.LinearRegression()
print(x)
# Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1), y)

# Veamos los coeficientes obtenidos, En nuestro caso, serán la Tangente
print('Coefficients: \n', regr.coef_)
# Este es el valor donde corta el eje Y (en X=0)
print('Independent term: \n', regr.intercept_)
# Error Cuadrado Medio
```

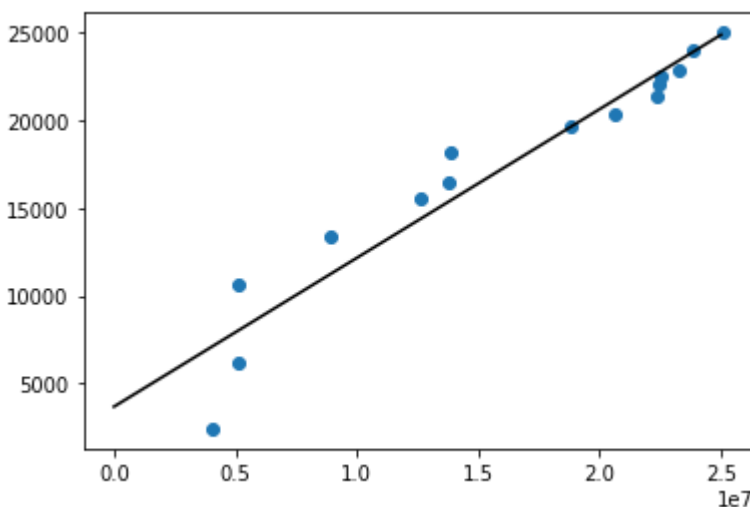
```
[4020319, 5087273, 5143156, 8883740, 12624324, 13806472, 13844086, 18798169,
20586005, 22373841, 22433379, 22489826, 23276191, 23848593, 25102469]
Coefficients:
[0.00084669]
Independent term:
3679.412891634178
```

In [73]:

```
#Graficar

plt.scatter(x, y)
print(y)
x_real = np.array(range(0, 25000000))
#print(x_real)
plt.plot(x_real, regr.predict(x_real.reshape(-1, 1)), color='black')
plt.show()
```

```
[2428, 6175, 10670, 13327, 15483, 16496, 18106, 19618, 20358, 21397, 22061,
22473, 22853, 23943, 24972]
```



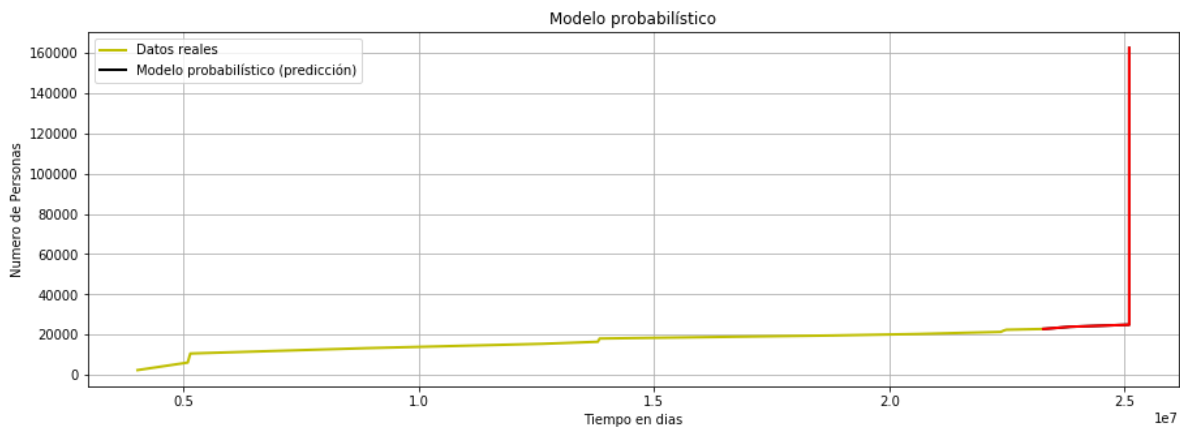
- Regresion Probabilistica

In [80]:

```
fig = plt.figure(figsize=(15,5))
ax = fig.add_subplot(111, axisbelow=True)
ax.plot(x[:len(x)-9], y[:len(x)-9], 'y', alpha=1, lw = 2, label = 'Datos reales')
ax.plot(x[len(x)-10:], y[len(x)-10:], 'black', alpha=1, lw = 2, label = 'Modelo probabilístico')
ax.set_xlabel('Tiempo en días')
ax.set_ylabel('Numero de Personas')
ax.set_title("Modelo probabilístico")
ax.legend()
ax.grid()
x_matematico = x[:]
y_matematico = y[:]
ax.plot(x[len(x)-10:], y[len(x)-10:], 'r', alpha=1, lw = 2, label = 'Modelo probabilístico (
```

Out[80]:

[<matplotlib.lines.Line2D at 0x19330589148>]



In [132]:

```
filtro = df1["tot"]
media = filtro.mean()
mediana = filtro.median()
prediccion = int(y[-1] + mediana)
print('_____')
print('-----MODELO PROBABILISTICO-----')
print('-PREDICCION ', prediccion)
print('-PREDICCION : ', y[20])
for i in range(x[-1], x[-1]+7):
    x.append(i)
    y.append(int(y[-1] + mediana))
print('_____')
print('SE ESTIMA QUE LASSO OBTENGA : ', y[-1]+7, 'DE VOTOS ')
```

```
-----MODELO PROBABILISTICO-----
-PREDICCION 4164370
-PREDICCION : 142680
```

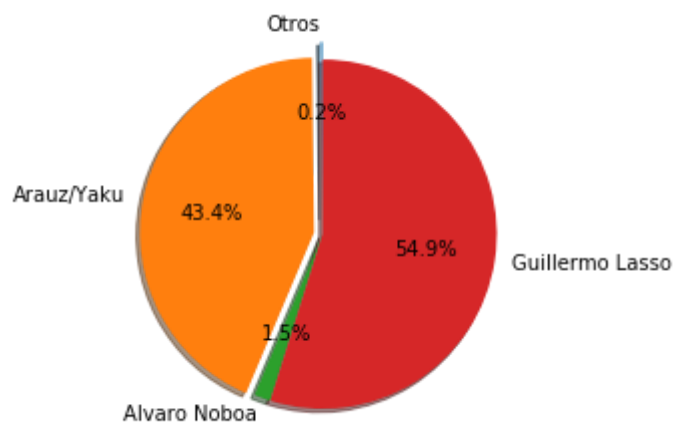
```
SE ESTIMA QUE LASSO OBTENGA : 4282085 DE VOTOS
```


In [131]:



```
labels = 'Otros', 'Arauz/Yaku', 'Alvaro Noboa', 'Guillermo Lasso'
sizes = [media, prediccion, y[20], x[1]]
explode = (0.1, 0.05, 0, 0) # only "explode" the 2nd slice (i.e. 'Hogs')

fig2, ax2 = plt.subplots()
ax2.pie(sizes, labels=labels, explode=explode, autopct='%1.1f%%', shadow=True, startangle=90)
plt.show()
```

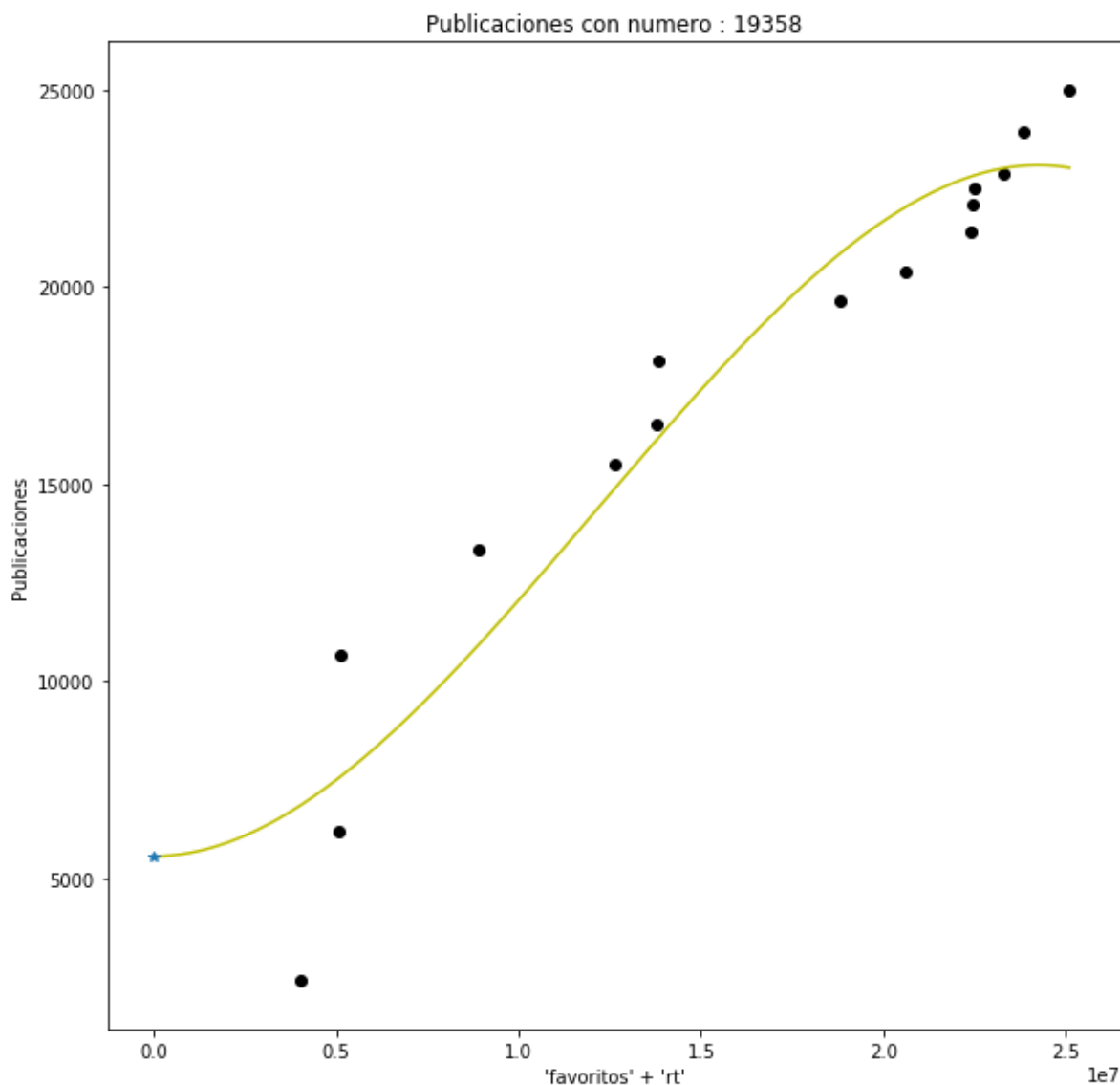


In [140]:

```
x = list(df2.iloc[:, 0]) # Estados
y = list(df2.iloc[:, 1]) # Total de la suma
z = api.get_user('LassoGuillermo').statuses_count

pf = PolynomialFeatures(degree = 3)
X = pf.fit_transform(np.array(x).reshape(-1, 1))
regresion_lineal = LinearRegression()
regresion_lineal.fit(X, y)
pred_x = list(range(0, max(x)+50))

puntos = pf.fit_transform(np.array(pred_x).reshape(-1, 1))
prediccion_entrenamiento = regresion_lineal.predict(puntos)
respuesta = round(-(prediccion_entrenamiento[z]))
plt.figure(figsize=(10,10))
plt.title('Publicaciones con numero : ' + str(z))
plt.plot(pred_x, prediccion_entrenamiento, color='y')
plt.scatter(x,y,label="Datos Reales",color="black")
plt.xlabel("'favoritos' + 'rt'")
plt.ylabel("Publicaciones")
plt.plot(z,prediccion_entrenamiento[z], '*')
plt.show()
```



Posteriormente se debe seguir un procesos de votación de eventos discretos que se describe a continuación: ◦ Tomar los resultados de la regresión para la selección del candidato. ◦ Se tiene una tendencia del 90% de personas que realizan el proceso de elección dentro del Ecuador. ◦ Dentro del procesos se tiene que alrededor del 5% - 10% votan nulo. ◦ Solo se va a tener en cuenta las elecciones de los asambleístas por el Azuay. ◦ Las personas solo tiene un recinto electoral para realizar el proceso. ◦ Las personas solo pueden realizar un proceso de elección por asambleísta del Azuay. ◦ La persona se acerca a la mesa electoral y hacen fila en caso de ser necesario. ◦ Realiza el voto en un tiempo aleatorio de un partido específico. ◦ La persona recibe su certificado votación

In [154]:



```
import simpy
import random
import matplotlib.pyplot as pp
import math
NUM_VOTANTES = 270
NUM_RECINTO = 1
NUM_ASAMBLEISTAS = 5
NUM_CANDIDATOS = 90
NUM_RECINTO = 1
NUM_MENSAS = 3
T_LLEGADAS = 20
TIEMPO_SIMULACION = 120
TOT_CLIENTES = 5

TIEMPO_ESPERA = 0.0 # tiempo de espera total
DURACION_TOTAL = 0.0 # duracion de servicio total
FIN = 0.0 # minuto en el que finaliza
```

In [171]:



```
now = datetime.now()
current_time = now.strftime("%H:%M:%S")
print("Current Time =", current_time)
llegada = lambda x, time : print(x, " llego en minuto: ", time, current_time) #Imprime Llegada
salida = lambda x, time : print(x, " Sale en minuto: ", time, current_time) #Imprime Salida
pasar_v = lambda x, time, esp : print(x, " pasa a vota en ", current_time, " minutos espera")
votacion = lambda x, time : print(x, " vota en ", time, current_time, " minutos") #Imprime Votacion

def votar(persona):
    global DURACION_TOTAL
    tiempo_votar = random.randint(1,5)
    yield env.timeout(tiempo_votar)
    votacion(persona, tiempo_votar)
    DURACION_TOTAL = DURACION_TOTAL + tiempo_votar # suma total de tiempo

def persona (env, name, mesas):
    global TIEMPO_ESPERA
    global FIN
    llega = env.now # minuto de Llegada de persona
    llegada(name, llega)
    with mesas.request() as request: #Esperamos pasar a mesas
        yield request # Obtiene turno
        pasa = env.now # Guarda el minuto cuando comienza a ser atendido
        espera = pasa - llega # Calcula el tiempo que espero
        TIEMPO_ESPERA = TIEMPO_ESPERA + espera # Acumula los tiempos de espera
        pasar_v(name, pasa, espera)
        yield env.process(votar(name)) # Invoca al de votar
        deja = env.now #Guarda el minuto en que termina el proceso votar
        salida(name, deja)
        FIN = deja # Conserva globalmente el ultimo minuto de la simulacion

def elecciones(env, personal):
    llegada = 0
    i = 0
    for i in range(TOT_CLIENTES): # Para n clientes
        R = random.random()
        llegada = -T_LLEGADAS * math.log(R) # Distribucion exponencial
        yield env.timeout(llegada) # Deja transcurrir un tiempo entre uno y otro
        print("-----")
        env.process(persona(env, 'Sufraga votante %d' % i, personal))

print ("***** Simulacion Elecciones *****")
random.seed (30) # Cualquier valor
env = simpy.Environment() # Crea el objeto entorno de simulacion
personal = simpy.Resource(env, NUM_MENSAS) #Crea las mesas
env.process(elecciones(env, personal)) #Invoca el proceso principal
env.run() #Inicia la simulacion
```

Current Time = 06:39:41

***** Simulacion Elecciones *****

```
-----
Sufraga votante 0 llego en minuto: 12.357767874449124 06:39:41
Sufraga votante 0 pasa a vota en 06:39:41 minutos esperando 3.0
Sufraga votante 0 vota en 1 06:39:41 minutos
Sufraga votante 0 Sale en minuto: 13.357767874449124 06:39:41
-----
```

Sufraga votante	1	llego en minuto:	37.17074959460619	06:39:41
Sufraga votante	1	pasa a vota en	06:39:41	minutos esperando 3.0
Sufraga votante	1	vota en	2	06:39:41 minutos
Sufraga votante	1	Sale en minuto:	39.17074959460619	06:39:41

Sufraga votante	2	llego en minuto:	46.67103921721893	06:39:41
Sufraga votante	2	pasa a vota en	06:39:41	minutos esperando 3.0

Sufraga votante	3	llego en minuto:	47.13787279405883	06:39:41
Sufraga votante	3	pasa a vota en	06:39:41	minutos esperando 3.0
Sufraga votante	2	vota en	1	06:39:41 minutos
Sufraga votante	2	Sale en minuto:	47.67103921721893	06:39:41
Sufraga votante	3	vota en	2	06:39:41 minutos
Sufraga votante	3	Sale en minuto:	49.13787279405883	06:39:41

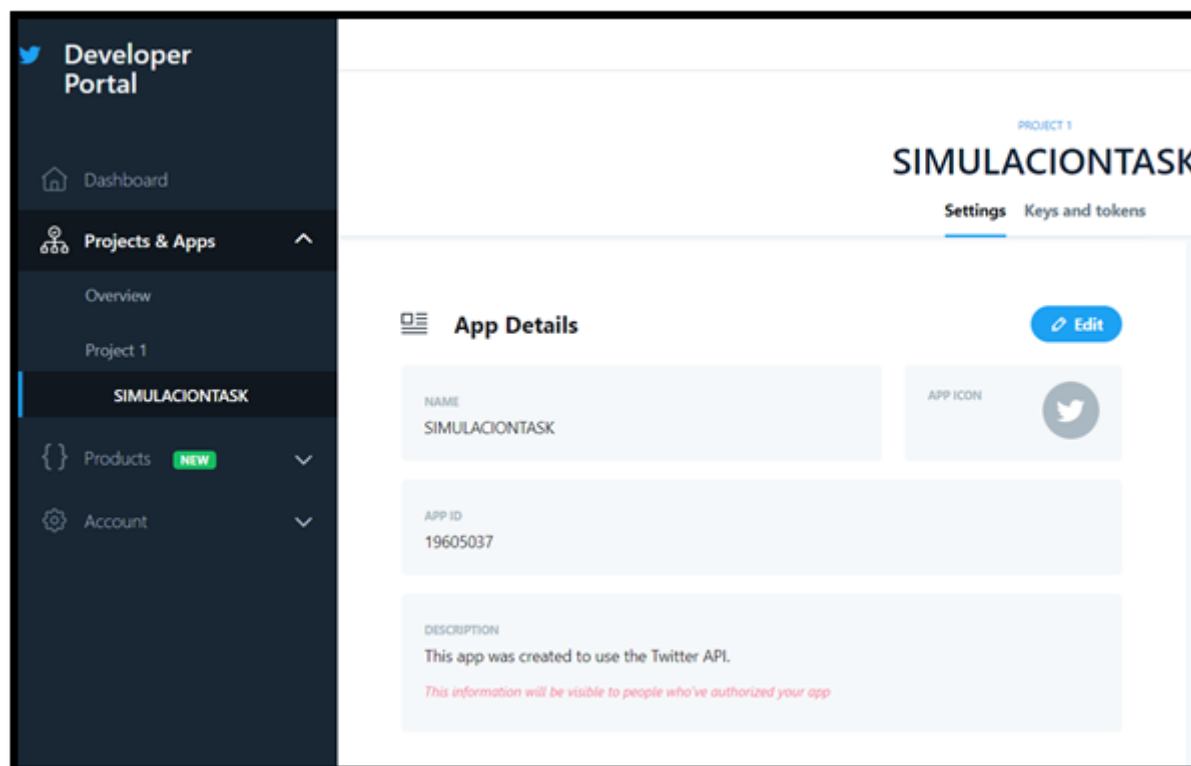
Sufraga votante	4	llego en minuto:	65.6042672111761	06:39:41
Sufraga votante	4	pasa a vota en	06:39:41	minutos esperando 3.0
Sufraga votante	4	vota en	1	06:39:41 minutos
Sufraga votante	4	Sale en minuto:	66.6042672111761	06:39:41

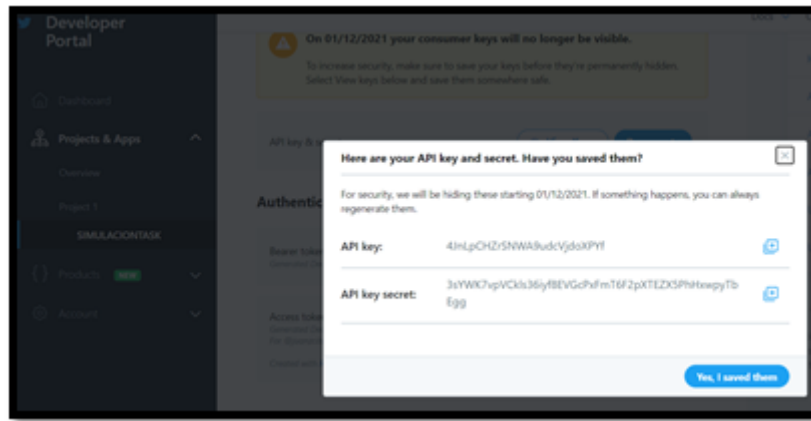


Conclusiones

Mediante esta practica usamos tweepy que es el api oficial de twitter, mediante el cual nos permite saber la informacion mas relevante de twitter, en este caso queriamos saber sobre los datos y estadisticas de los candidatos mas populares, para poder realizar un analisis, a ver que candidato es el mas popular en dicha red.

Anexos:





Recomendaciones

Tener instalado la libreria tweepy, mediante la cual consumimos datos de twitter

Librerías como sympy que nos permite simular las votaciones

Referencias: [1]: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/>
(<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/>) [2]:
<https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>
(<https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/>) [3]:
<https://eprints.ucm.es/48804/1/TFM%20Manuel%20Alejandro%20Rodriguez%20Santana.pdf>
(<https://eprints.ucm.es/48804/1/TFM%20Manuel%20Alejandro%20Rodriguez%20Santana.pdf>) [4]:
http://opac.pucv.cl/pucv_txt/txt-8000/UCC8094_01.pdf (http://opac.pucv.cl/pucv_txt/txt-8000/UCC8094_01.pdf)