

# EXAMEN INTERCICLO

Nombre: David Leon

## SIMULACION

```
In [1]: # instalamos Librerias
import simpy
import math
import random
import tweepy
import numpy as np
import random
import matplotlib.pyplot as plt
import json
from termcolor import colored
from string import digits
import pandas as pd
```

```
In [2]: from requests import ReadTimeout, ConnectTimeout, HTTPError, Timeout, ConnectionError
```

```
In [3]: #OBTENCION DE DATOS DE TWITER
consumer_key = "L6XdJhLezyZ98nm5mZIkHMYEn"
consumer_secret = "cNs8bmW62QujCnJvKRaqXk98Ghf97VASfvVqQeaPyms0AGarHw"
access_token = "1334387936847155200-1jrvfm9bnRV9nieDpk6c44IS8JnCh"
access_token_secret = "78e15Mctcvf116DDIVLsueKq08duintbb4CmrFTttu08an"
```

```
auth = tweepy.OAuthHandler(consumer_key,consumer_secret)
auth.set_access_token(access_token,access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)

title = []
word_count = []
links = []
hashtags = []
user_mentions = []
favorite_count = []
retweet_count = []
cont = 1
conts = 0
language = 'es'
try :

    for tweet in tweepy.Cursor(api.search, q="Guillermo Lasso", tweet_mode="extended",lang=language).items(5000):
        if (not tweet.retweeted) and ('RT @' not in tweet.full_text):

            if cont <= 2500:

                # Title
                #titl = tweet.full_text
                title.append(tweet.full_text)
                # Word count
                cadenaPalabras = str(tweet.full_text)
                listaPalabras = cadenaPalabras.split()
                word_count.append(len(listaPalabras))
                # Link
                links.append(len(tweet.entities['urls']))
                # Hashtags
                hashtags.append(len(tweet.entities['hashtags']))
                # Menciones Usuarios
                hashtags.append(len(tweet.entities['hashtags']))
                # Menciones Usuarios
                user_mentions.append(len(tweet.entities['user_mentions']))
                # Me gusta
                favorite_count.append(tweet.favorite_count)
                # Retweet
                retweet_count.append(tweet.retweet_count)

                print(cont, '.',
                    colored("-- Texto -- ", 'red'), tweet.full_text,
                    colored("\n -- Me gusta--", 'red'), tweet.favorite_count,
                    colored("-- Retweet --", 'red'), tweet.retweet_count,
                    colored("\n -- Link -- ", 'red'), tweet.entities['urls'],
                    colored("\n -- Menciones Usuarios ---- ", 'red'), tweet.entities['user_mentions'],
                    colored("\n -- Hashtags -- ", 'red'), tweet.entities['hashtags'],
                    colored("\n -- Numero de Palabras -- ", 'red'), len(listaPalabras),
                    colored("\n -- Cantidad de Menciones -- ", 'red'), len(tweet.entities['user_mentions']),
                    colored("\n -- Cantidad de Links -- ", 'red'), len(tweet.entities['urls']),
                    colored("\n -- Cantidad de hashtags -- ", 'red'), len(tweet.entities['hashtags']), '\n')

                cont = cont+1

            else:
                print('Llego a las 2500 Datos')
                break;

except TimeoutError:
    print('TimeoutError')
except NewConnectionError:
    print('NewConnectionError')
except MaxRetryError:
```

```

        print('MaxRetryError')
    except NewConnectionError:
        print('ConnectionError')
    except TweepError:
        print('TweepError')
    except UBICACIONDoesNotExist:
        print('UBICACIONDoesNotExist')
    except NameError:
        print('NameError')

40 , display_url : twitter.com/revistavistazo... , indices : [232, 255]]
-- Menciones Usuarios ---- [{"screen_name": 'LeonidasIzaSal1', 'name': 'Leonidas Iza Salazar Oficial', 'id': 11877193568825
26209, 'id_str': '1187719356882526209', 'indices': [0, 16]}]
-- Hashtags -- []
-- Numero de Palabras -- 41 -- Cantidad de Menciones -- 1
-- Cantidad de Links -- 1 -- Cantidad de hashtags -- 0

836 . -- Texto -- Qué nos invada y colonice Ecuador con su presidente liberal, Guillermo Lasso.

Fin.
-- Me gusta-- 9 -- Retweet -- 0
-- Link -- []
-- Menciones Usuarios ---- []
-- Hashtags -- []
-- Numero de Palabras -- 13 -- Cantidad de Menciones -- 0
-- Cantidad de Links -- 0 -- Cantidad de hashtags -- 0

837 . -- Texto -- Saludo protocolar del Presidente Guillermo Lasso junto al Ministro de Relaciones Exteriores Embajador Maur
icio Montalvo al Cuerpo Diplomático acreditado en el Ecuador. https://t.co/IUjKh1fwjI
-- Me gusta-- 29 -- Retweet -- 3

```

```

In [4]: dict = {'Title':title, 'Word_Count':word_count, 'Links':links, 'Hashtags':hashtags,
              'User_Mentions':user_mentions, 'Favorite':favorite_count, 'Retweet':retweet_count}
df = pd.DataFrame(dict)
df.to_csv('salida.csv')

```

```

In [6]: result = {'Title':title, 'Word_Count':word_count, 'Links':links, 'Hashtags':hashtags,
                'User_Mentions':user_mentions, 'Favorite':favorite_count, 'Retweet':retweet_count}
respaldo = result
df = pd.DataFrame(data = result)
df.head(20)

```

```

Out[6]:

```

	Title	Word_Count	Links	Hashtags	User_Mentions	Favorite	Retweet
0	Siguiente por favor. \n\n <a href="https://t.co/FnlEpHPB...">https://t.co/FnlEpHPB...</a>	5	2	0	0	0	0
1	El presidente de Ecuador, Guillermo Lasso, anu...	32	0	0	0	1	0
2	@sukamonsalve @nessateran Es la traducción de ...	8	0	0	2	0	0
3	@ing_carlos87 @LorenPAvarez Sufran correistas...	19	0	0	2	0	0
4	Si Guillermo Lasso cumple su plan de vacunaci...	19	0	0	0	1	0
5	El fascista Guillermo Lasso amenazó a grupos s...	48	1	1	0	0	0
6	Guillermo Lasso dice que no habrá sabatinas po...	48	0	1	0	0	0
7	@CarlosVerareal Si en el gabinete de Guillermo...	19	0	0	1	0	0
8	@BNPeriodismo Pasaron ahora a servir a Guiller...	12	0	0	1	0	0
9	LEONIDAS IZA A GUILLERMO LASSO: PARE CON EL FM...	18	1	0	1	0	0
10	🇪🇺 FASCISTA LASSO, ENVÍA MENSAJE DE APOYO A FASC...	41	0	3	0	2	1
11	CONOZCA al Verdadero GUILLERMO LASSO JURA "MA...	16	1	0	1	1	1
12	@APACHE_EC El gobierno de Guillermo Lasso ha s...	40	0	0	1	1	0
13	Ecuador: Guillermo Lasso participará en confer...	11	1	0	0	0	0
14	Lasso participará en conferencia sobre migrant...	8	1	0	0	0	1
15	El presidente de Ecuador Guillermo Lasso recib...	15	1	0	0	0	0
16	🇪🇺 Consulta tu lugar de vacuna\n\nEl president...	37	1	0	0	0	0
17	El presidente de Ecuador Guillermo Lasso recib...	15	1	0	0	0	0
18	El presidente de Ecuador Guillermo Lasso recib...	14	1	0	0	0	0
19	Daniel_RiosVE: El presidente de #Ecuador Guill...	43	0	2	0	0	0

```

[7]: df.describe()

```

```

[7]:

```

	Word_Count	Links	Hashtags	User_Mentions	Favorite	Retweet
count	933.000000	933.000000	933.000000	933.000000	933.000000	933.000000
mean	31.382637	0.370847	0.491961	0.831726	12.998928	4.111468
std	12.479356	0.500737	1.147487	1.319457	64.361447	19.208770
min	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	20.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	33.000000	0.000000	0.000000	0.000000	0.000000	0.000000

	Stu	12.473330	0.000737	1.147407	1.312497	04.301447	13.200770
min	4.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	20.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	33.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	42.000000	1.000000	1.000000	1.000000	3.000000	1.000000	1.000000
max	59.000000	2.000000	11.000000	13.000000	1024.000000	370.000000	

## REGRESION LINEAL

```
15]: from sklearn.linear_model import LinearRegression
from sklearn import linear_model

word_Count = df.Word_Count

x = range(1,len(word_Count)+1)

# Hacemos las predicciones que en definitiva una línea (en este caso, al ser 2D)
y = df.Retweet

regr = linear_model.LinearRegression()
# Entrenamos nuestro modelo
regr.fit(np.array(x).reshape(-1, 1),y)

y_prediccion = regr.predict([[64]])

x_real = np.array(range(0, len(x)))
```

**Posteriormente se debe seguir un procesos de vacunación en los recintos electorales que se describe a continuación (Tomar el proceso para el proyecto final):**

- Solo se va a tener en cuenta uno de los recintos electorales (investigar datos de cuantas personas asisten a votar).

AZUAY > CUENCA > EL VECINO > EL VECINO			
Actas válidas (112)		100,00%	
Actas con novedad (0)		0,00%	
Total (112 de 112) Actas		100,00%	
		SUFRAGANTES	
			Total
		Sufragantes	31331
		Ausentismo	7835
		VER REPORTE	

```
In [16]: sufragantes =31331
ausentismo =7835
votar_personas = sufragantes + ausentismo
print(votar_personas)
```

```
In [16]: sufragantes =31331
ausentismo =7835
votar_personas = sufragantes + ausentismo
print(votar_personas)

39166
```

```
In [18]: #Cuantas vacunas deben realizarse en un mes para Lograr vacunar a todas Las personas
vacunas_dia = votar_personas/(29)
print(vacunas_dia)

1350.551724137931
```

**Se obtiene que en el mes el 80% realizaron la vacuna**

```
In [19]: promedio80 = (votar_personas * 0.8)
print("80% de personas: ", promedio80)

80% de personas: 31332.800000000003
```

**Dentro del procesos se tiene que alrededor del 5% - 10% no podrán vacunarse.**

```
In [20]: promedio10 = (votar_personas * 0.1)
print("10% de personas: ", promedio10)

10% de personas: 3916.6000000000004
```

### Las personas solo tiene un recinto electoral para realizar el proceso.

- Las personas realizan la primera vacuna y 30 días después la segunda vacuna.
- La persona se acerca a la mesa y hacen fila en caso de ser necesario para recibir la vacuna.
- Realiza la vacunación en un tiempo aleatorio entre 5 a 10 minutos.

```
In [21]: segunda_dosis_fecha = 30

tiempo_vacunacion = random.randint(5, 10)
ab = 10
for a in range(10):
    a = random.randint(5, 10)
    print(a)
print(tiempo_vacunacion)

10
8
5
10
8
5
5
7
7
7
10
```

### Debe esperar 20 minutos dentro del establecimiento para verificar que no tenga problemas de salud.

- La persona recibe su certificado de vacunación y la fecha de la próxima vacuna entre 2 – 3 minutos.
- RECUERDA EL HORARIO DE ATENCION EN EL CENTRO DE VACUNACION ES DE 8:00 A 17:00

```
[26]: tiempo_verificar = 20

tiempo_vacunacion_total = tiempo_vacunacion + tiempo_verificar
print(tiempo_vacunacion_total)
# El lugar de vacunacion tiene horario de atencion de 9 horas, en total 540 minutos
horario_atencion = 60 * (9)
print(horario_atencion)

# Días de vacuna
horario_atencion_completa = horario_atencion * 31
print(horario_atencion_completa)

personas_vacunar = horario_atencion/tiempo_vacunacion_total
personas_vacunar_dia = votar_personas/30
print(personas_vacunar)
print(personas_vacunar_dia)

30
540
16740
18.0
1305.5333333333333
```

```
[38]: intervalo_llegada = horario_atencion/vacunas_dia
print(intervalo_llegada, "en minutos son ", 3)
```

```
print(intervalo_llegada, "en minutos son ", 3)
intervalo_llegada = 3
```

0.3998365929632845 en minutos son 3

```
n [45]: import simpy
import random
from random import choice
import matplotlib.pyplot as pp

%matplotlib inline
```

```
n [46]: # Maximo de vacunas
MAX_VACUNAS = 1350
# Total de numero de puntos de vacunacion
JUNTAS = 118
PACIENTES = 39166
# Intervalo de tiempo en que llegan los pacientes
INTERVALO_LLEGADA = 5 #minutos
# Tiempo de simulación
TIEMPO_SIMULACION = 775

# Creamos un diccionario para almacenar las horas en que se vacuna los pacientes
tiempo_vacunacion={}
#Diccionarios para almacenar la primera y segunda dosis
pri_dosis = {} #
seg_dosis = {} #
#Diccionario para el numero de pacientes que no han sido atendidos
p_no_vacunado = []

#Numero de vacunas aplicadas
n_vacunas_apli = 0 #
```

```

n_vacunas_ap11 = 0 #
#Fecha inicial 30 días, y se suma 1 día.
fecha_inicial =30

class Recinto(object):
    def __init__(self, environment, num_mesas, nombre_recinto):
        # Guardamos como variable el entorno de ejecucion
        self.env=environment
        # Creamos el recurso que representa las mesas o juntas
        self.mesas = simpy.Resource(environment, num_mesas)
        self.nombre_recinto = nombre_recinto

        # Segun UNICEF las personas que no se beberian vacunar son las que presentan alergias a los componentes
        # de las vacunas y las que presenten sintomas de COVID-19
        # Para validar si un paciente tiene sintomas o no haremos
        def verificacion_sintomas(self, paciente, tam_no_vacunados, n_per_sintomas):
            estado = choice(["con sintomas", "sin sintomas"])
            if tam_no_vacunados<=n_per_sintomas and estado == "con sintomas":
                p_no_vacunado.append(paciente)
            return estado

        #La persona se acerca y hace fila para recibir la vacunacion de la primera dosis
        def pri_dosis(self, paciente, sintomas):
            if sintomas == "sin sintomas":
                t_vacunacion = random.randint(5,10) # ==Realiza la vacunación en un tiempo aleatorio entre 5 a 10 minutos.
                yield self.env.timeout(t_vacunacion)
                print("El paciente ", paciente, " ha sido vacunado con la primera dosis.")
                print("La aplicacion de la segunda dosis es luego de 30 días")

                pri_dosis[paciente] = t_vacunacion
            else:
                print("El paciente ",paciente," presenta sintomas de COVID-19, por lo que no puede ser vacunado.")

```

```

def seg_dosis(self, paciente, sintomas):
    if sintomas == "sin sintomas":
        t_vacunacion = random.randint(5,10)
        yield self.env.timeout(t_vacunacion)
        print("El paciente ",paciente, "ha sido vacunado con la segunda dosis.")
        seg_dosis[paciente] = t_vacunacion
    else:
        print("El paciente ",paciente," tiene sintomas, por lo que no puede ser vacunado")

#Debe esperar 20 minutos dentro del establecimiento para verificar que no tenga problemas de salud.
def postverificacion_salud(self,paciente):
    post_t_espera = 20
    yield self.env.timeout(post_t_espera)

#La persona recibe su certificado de vacunación y la fecha de la próxima vacuna entre 2 - 3 minutos.
def generar_certificado(self,paciente):
    t_salida = random.randint(2,3)
    yield self.env.timeout(t_salida)

def llegada(env, paciente, punto, tam_no_vacunados, n_per_sintomas):
    # Usamos el reloj de la simulacion (env.now()) para indicar a la
    # hora que llega el vehiculo con el nombre pasado como parametro
    print('Llega el paciente: %s a la hora %.2f.' % (paciente, env.now))

    # Especificamos que vamos a usar un recurso (Resource) que representa
    with punto.mesas.request() as mesa:
        # Ocupamos la mesa de vacunacion
        yield mesa

    #Indicamos que el paciente se esta trasladando para ser vacunado
    print('Se traslada el paciente: %s a la hora %.2f.' % (paciente,env.now))

    #yield env.process(punto.examen_sintomas(paciente.tam no vacunados, n per sintomas))

```

```

print('Se traslada el paciente: %s a la hora %.2f.' % (paciente,env.now))

#yield env.process(punto.examen_sintomas(paciente,tam_no_vacunados, n_per_sintomas))
sintomas = punto.verificacion_sintomas(paciente, tam_no_vacunados, n_per_sintomas)

print("Sintomas: ",sintomas)
if sintomas == "sin sintomas":
    yield env.process(punto.pri_dosis(paciente, sintomas))
    print("El paciente %s esta en proceso para la primera dosis a las %.2f." \
          %(paciente,env.now))

    yield env.process(punto.postverificacion_salud(paciente))
    print("Se espero 20 min dentro del establecimiento desde las %.2f. \
          para el paciente %s" \
          %(env.now,paciente))

    yield env.process(punto.generar_certificado(paciente))
    print("Se ha generado el certificado y fecha para la segunda dosis a las %.2f. \
          para el paciente %s" \
          %(env.now,paciente))
if sintomas == "sin sintomas":
    yield env.process(punto.seg_dosis(paciente, sintomas))
    print("El paciente %s esta en proceso para la segunda dosis a las %.2f." \
          %(paciente,env.now))

def simulacion(env, num_mesas, pacientes, intervalo):
    recinto = Recinto(env, num_mesas, "Universidad de Cuenca")
    #Generado randomico porcentaje entre 5% y 10% con sintomas
    p_sintomas = random.randint(5,10)
    print("Porcentaje de personas que tienen sintomas: ",p_sintomas,"%")

    #Calculado numero de personas con sintomas
    n_per_sintomas = int(pacientes/100*p_sintomas)

```

```

    #Calculado numero de personas con sintomas
    n_per_sintomas = int(pacientes/100*p_sintomas)
    print("Numero de personas que tienen sintomas: ",n_per_sintomas)

    #Numero de personas no vacunadas
    tam_no_vacunados = len(p_no_vacunado)

    for i in range(pacientes):
        env.process(llegada(
            env,'Paciente-%d'%(i+1),recinto,tam_no_vacunados,n_per_sintomas))
    while True:

        yield env.timeout(random.randint(intervalo-2,intervalo+2))
        i+=1
        env.process(llegada(env,'Paciente-%d'%(i+1),recinto,tam_no_vacunados,n_per_sintomas))

```

```

[48]: # Creamos el entorno de simulacion
env=simpy.Environment()
print("# de Pacientes => ", PACIENTES)
env.process(simulacion(env, JUNTAS, PACIENTES, INTERVALO_LLEGADA))
# Ejecutamos el proceso durante el tiempo de simulacion
env.run(until = TIEMPO_SIMULACION)

Se ha generado el certificado y fecha para la segunda dosis a las 144.00.
Se ha generado el certificado y fecha para la segunda dosis a las 144.00.
Se ha generado el certificado y fecha para la segunda dosis a las 144.00.
Se traslada el paciente: Paciente-932 a la hora 144.00.
Sintomas: con sintomas
Se traslada el paciente: Paciente-933 a la hora 144.00.
Sintomas: con sintomas
Se traslada el paciente: Paciente-934 a la hora 144.00.
Sintomas: sin sintomas
Se traslada el paciente: Paciente-935 a la hora 144.00.

Se traslada el paciente: Paciente-936 a la hora 144.00.
Sintomas: sin sintomas
Se traslada el paciente: Paciente-940 a la hora 144.00.

```

```

In [58]: person_vacunas = vacunados
tam_no_vacunados = len(p_no_vacunado)

print("NUMERO TOTAL NO VACUNADOS",tam_no_vacunados)

tot_vacunados=PACIENTES-tam_no_vacunados
print("TOTAL NUMEROS VACUNADOS",tot_vacunados)

NUMERO TOTAL NO VACUNADOS 4949
TOTAL NUMEROS VACUNADOS 34217

```

```

In [59]: no = tam_no_vacunados
si = tot_vacunados
plt.figure(figsize=(8,5))

etiquetas = ['No vacunados', 'Vacunados']
colores = ['#a0522d', '#aabbcc']
plt.pie([no, si], labels=etiquetas, colors=colores, autopct='%.2f %%')
plt.title('Vacunación')
plt.show()

```

Vacunación

```
plt.show()
```

Vacunación

