

Nombre: Alex Reinoso

Materia: Simulacion

Tema: Examen final practico

Fecha: 02/26/2022

▼ Instalar

```
pip install simpy
```

```
Requirement already satisfied: simpy in /usr/local/lib/python3.7/dist-packages (4.0.1)
```

▼ Investigacion

Cantidad de turistas que visitaron Cuenca en el feriado de noviembre de 2021: 90.000

Cantidad de turistas que visitaron Manta: 100.000,00

Cantidad de turisistas que visitaron Tungurahua: 20.000,00 la mayoría desde GYE y UIO

- Banos: 8.000,00 plazas en el hospedaje
- Ambato: 4.500,00

Fuente: <https://dialoguemos.ec/2021/11/sector-turistico-del-pais-se-reactiva-tras-el-feriado/>

▼ Importaciones

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import stats
import seaborn as sns
import collections
import random
import simpy
from numpy.core.fromnumeric import size
from scipy.stats import bernoulli, binom
```

▼ Variables

```
CAPACIDAD = 40
NUM_BUSES=10
TIEMPO_SIMULACION = 96 #son 4 dias de feriado y esto hace 96 horas

MEDIA=0.16
DESV_S=0.03
NOMBRES = ['Arnold', 'Charles', 'Tupac','Nicki','Estefania','Alan','Ronald','Nadia','Esteban',
```

▼ Funciones

```
def generarGenero(n):
    p = 0.37
    res = []
    for i in range(0,n):
        res += randomBernoulliVariables(p)
    return res
```

```
def generarNombre(n):
    res = []
    for i in range(0, n):
        rn = random.randint(0,9)
        res.append(NOMBRES[rn])
    return res
```

```
def randomBernoulliVariables(p):
    if np.random.rand() <= p:
        return 'M'
    else:
        return 'F'
```

```
def poissonEdades(s):
    rng = np.random.default_rng()
    arr = rng.poisson(20,s)
    arr = arr[arr > 18]
    if (arr.size < s):
        ar2 = rng.poisson(20,s)
        ar2 = ar2[ar2 > 18]
        res = np.concatenate([arr, ar2])
        if (res.size < s):
            poissonEdades(s)
        return res[0:s]
    else:
        return res[0:s]
```

```

else:
    return arr[0:s]

```

▼ Simulacion

```

import collections
import random
import simpy

def empresa(env, empresaturismo, destino):
    with empresaturismo.recurso_buses.request() as turno:
        yield turno
        imprimir_datos_recurso_bus(recurso_buses) # Informacion de los buses
        print('Bus con destino: %s llegara en %.1f horas' %(destino, empresaturismo.hours_desti[destino])
        yield env.timeout(empresaturismo.hours_desti[destino])
        empresaturismo.recurso_buses.release(turno)
        print('Bus llego a : %s en %.1f horas' %(destino, empresaturismo.hours_desti[destino]))
        print('Se libera el bus')

def imprimir_datos_recurso_bus(rec):
    print(f'{rec.count} de {rec.capacity}')

def ventaBoletos(env, num_pasajes, destino, empresaturismo):
    with empresaturismo.recurso_cout_pasajes.request() as turno:
        resultado = yield turno | empresaturismo.sold_out[destino]
        if empresaturismo.num_pasajes[destino] < num_pasajes:
            yield env.timeout(0.5)
            return
        empresaturismo.num_pasajes[destino] -= num_pasajes
        if empresaturismo.num_pasajes[destino] == 0:
            empresaturismo.num_buses[destino] += 1
            env.process(empresa(env, empresaturismo, destino))
            empresaturismo.num_pasajes[destino] = 40
        visitados[destino] += 1

def llegadaPasajeros(env, empresaturismo):
    while True:
        normal = stats.norm(MEDIA, DESV_S)
        yield env.timeout(normal.rvs(1)) #tiempo de llegado estimado de los pasajeros
        destino = random.choices(empresaturismo.destinos, empresaturismo.probabilidad, k=1)#se se
        num_pasajes = 1
        cantidad = 1#esto es para generar la persona
        generos = generarGenero(cantidad)
        edades = poissonEdades(cantidad)
        nombres = generarNombre(cantidad)
        rslt = zip(nombres, edades, generos)
        res = list(rslt)
        print('Cliente %s con destino a %s' %(res, destino))
        if empresaturismo.num_pasajes[destino][0]:
            env.process(ventaBoletos(env, num_pasajes, destino[0], empresaturismo))

```

```

ABCTURISMO = collections.namedtuple('ABCTURISMO', 'recurso_cout_pasajes, recurso_buses, hours_
env = simpy.Environment()

recurso_cout_pasajes = simpy.Resource(env, capacity=1)
recurso_buses = simpy.Resource(env, capacity=NUM_BUSES)
DESTINOS = ['MANTA', 'GYE', 'BANOS', 'AMBATO']
horas_viaje = [11, 5, 7, 9]
probabilidad = [0.45, 0.20, 0.15, 0.20]
hours_desti=dict(zip(DESTINOS, horas_viaje))
num_pasajes = {destino: CAPACIDAD for destino in DESTINOS}
sold_out = {destino: env.event() for destino in DESTINOS}
num_buses = {destino: 0 for destino in DESTINOS}
visitados = {destino: 0 for destino in DESTINOS}

abcturismo = ABCTURISMO (recurso_cout_pasajes, recurso_buses, hours_desti, probabilidad, DES
env.process(llegadaPasajeros(env, abcturismo))
env.run(until=TIEMPO_SIMULACION)#Aqui le mando el tiempo de 4 dias

print('REPORTE ABC-TURISMO')
for destino in DESTINOS:
    if abcturismo.sold_out[destino]:
        print('Destino: %s necesita %.1f buses' %(destino, abcturismo.num_buses[destino]))
        Cliente [] con destino a ['MANTA']
        Cliente [('Pablo', 25, 'F')] con destino a ['AMBATO']
        Cliente [('Esteban', 19, 'M')] con destino a ['GYE']
        Cliente [('Arnold', 22, 'F')] con destino a ['GYE']
        Cliente [('Ronal', 22, 'M')] con destino a ['AMBATO']

        Cliente [('Nadia', 19, 'F')] con destino a ['MANTA']
        Cliente [('Ronal', 24, 'F')] con destino a ['MANTA']
        Cliente [('Esteban', 19, 'F')] con destino a ['MANTA']
        Cliente [('Arnold', 20, 'F')] con destino a ['GYE']
        Cliente [] con destino a ['AMBATO']
        Cliente [('Pablo', 22, 'M')] con destino a ['BANOS']
        Cliente [('Arnold', 19, 'M')] con destino a ['AMBATO']
        Cliente [('Arnold', 20, 'F')] con destino a ['MANTA']
        Cliente [('Nicki', 28, 'M')] con destino a ['GYE']
        Cliente [('Charles', 20, 'F')] con destino a ['AMBATO']
        Cliente [('Estefania', 20, 'M')] con destino a ['MANTA']
        Cliente [] con destino a ['AMBATO']
        Cliente [('Pablo', 19, 'M')] con destino a ['GYE']
        Cliente [] con destino a ['MANTA']
        Cliente [('Pablo', 21, 'F')] con destino a ['MANTA']
        Cliente [('Arnold', 19, 'M')] con destino a ['MANTA']
        Cliente [('Tupac', 23, 'M')] con destino a ['BANOS']
        Cliente [('Arnold', 20, 'F')] con destino a ['MANTA']
        Bus llego a : MANTA en 11.0 horas
        Se libera el bus
        Cliente [('Tupac', 21, 'F')] con destino a ['AMBATO']
        Cliente [('Arnold', 26, 'F')] con destino a ['MANTA']
        Cliente [] con destino a ['BANOS']
        1 de 10
        Bus con destino: BANOS llegara en 7.0 horas
        Cliente [('Nicki', 24, 'M')] con destino a ['MANTA']

```

```

Cliente [( 'NICKI' , 24, 'M' )] con destino a [ 'MANTA' ]
Cliente [( 'Esteban', 20, 'M')] con destino a [ 'GYE' ]
Cliente [] con destino a [ 'GYE' ]
Cliente [( 'Nadia', 22, 'F')] con destino a [ 'MANTA' ]
Cliente [( 'Tupac', 28, 'F')] con destino a [ 'MANTA' ]
Cliente [] con destino a [ 'GYE' ]
Cliente [] con destino a [ 'MANTA' ]
Cliente [] con destino a [ 'AMBATO' ]
Cliente [( 'Pablo', 22, 'F')] con destino a [ 'GYE' ]
Cliente [( 'Estefania', 24, 'F')] con destino a [ 'GYE' ]
Cliente [( 'Esteban', 22, 'M')] con destino a [ 'BANOS' ]
Cliente [( 'Charles', 20, 'F')] con destino a [ 'MANTA' ]
Cliente [] con destino a [ 'BANOS' ]
Cliente [( 'Nicki', 20, 'F')] con destino a [ 'GYE' ]
Cliente [( 'Ronal', 24, 'M')] con destino a [ 'MANTA' ]
Cliente [] con destino a [ 'GYE' ]
2 de 10
Bus con destino: GYE llegara en 5.0 horas
Cliente [( 'Estefania', 28, 'M')] con destino a [ 'GYE' ]
Cliente [( 'Ronal', 19, 'M')] con destino a [ 'GYE' ]
Cliente [] con destino a [ 'AMBATO' ]
Cliente [( 'Tupac', 25, 'M')] con destino a [ 'BANOS' ]
Cliente [] con destino a [ 'MANTA' ]
REPORTE ABC-TURISMO
Destino: MANTA necesita 6.0 buses
Destino: GYE necesita 3.0 buses
Destino: BANOS necesita 2.0 buses
Destino: AMBATO necesita 2.0 buses

```

```

plt.pie(list(visitados.values()), labels=list(visitados.keys()))
plt.title('Destinos mas visitados')
plt.show()

```

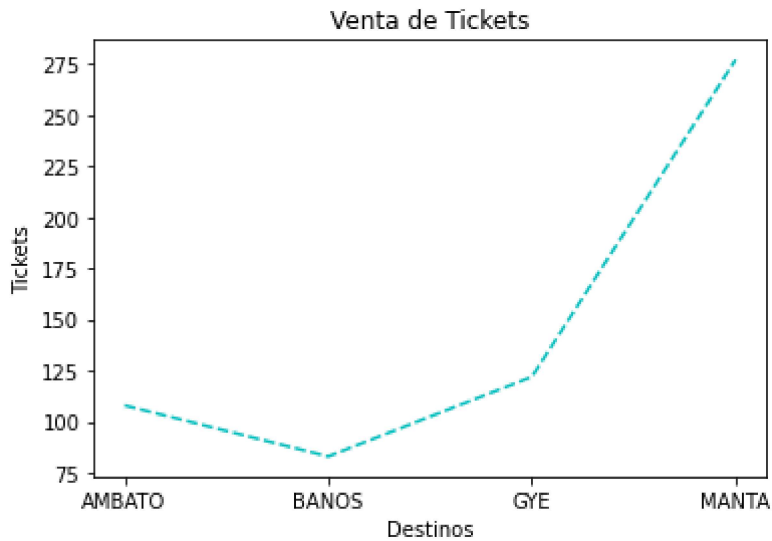


```

data = visitados.items()
data = sorted(data)
x, y = zip(*data)
plt.plot(x, y, 'c--')
plt.title('Venta de Tickets')
plt.xlabel('Destinos')

```

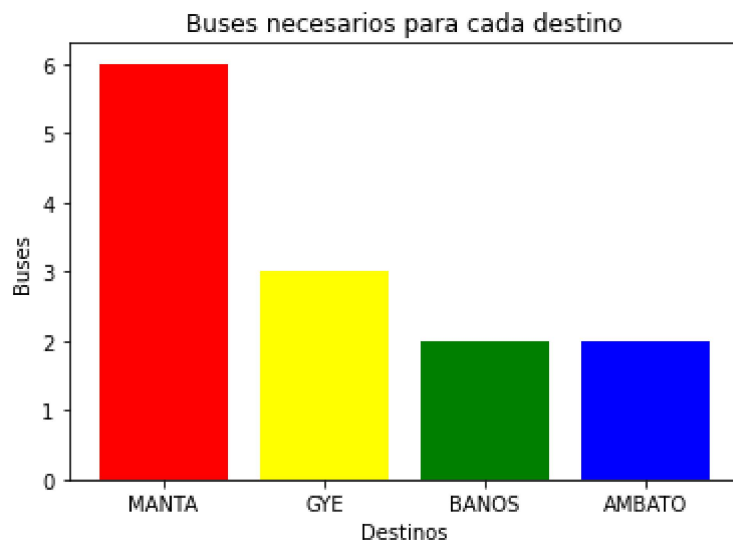
```
plt.ylabel('Tickets')  
plt.show()
```



visitados

```
{'AMBATO': 135, 'BANOS': 82, 'GYE': 112, 'MANTA': 269}
```

```
etiqueta = list(num_buses.keys())  
valores = list(num_buses.values())  
plt.bar(range(len(num_buses)), valores, align="center", tick_label=etiqueta, color=['red', 'y'  
plt.title("Buses necesarios para cada destino")  
plt.xlabel("Destinos")  
plt.ylabel("Buses")  
plt.show()
```



✓ 0 s completado a las 21:49

● ✕