

In [1]: `!pip install simpy`

```
Collecting simpy
  Downloading https://files.pythonhosted.org/packages/20/f9/874b0b
ab83406827db93292a5bbe5acb5c18e3cea665b2f6e053292cb687/simpy-4.0.1
-py2.py3-none-any.whl (https://files.pythonhosted.org/packages/20/
f9/874b0bab83406827db93292a5bbe5acb5c18e3cea665b2f6e053292cb687/si
mpy-4.0.1-py2.py3-none-any.whl)
Installing collected packages: simpy
Successfully installed simpy-4.0.1
```

Películas/Cine

Tipicos:

- Recursos
- Condición de eventos
- Eventos compartidos

Enunciado:

Modelar una sala de cine con un mostrador para la venta de entradas a tres tipos de películas diferentes, tener en cuenta que se modela solo para un espectáculo.

- Las personas llegan en momentos aleatorios e intentan comprar un número aleatorio de boletos (1-6) para una película al azar.
- Cuando se agota las entradas a una película, las personas que esperan por un boleto salen de la cola.
- La sala de cine es un Contenedor que abarca todos los procesos.
- El mostrador es un Recurso con la capacidad de uno.
- Finalmente, si la última persona desea comprar más boletos de los existentes se le vende solo los que tiene el mostrador.

```
In [20]: 1 import collections
          2 import random
          3 import simpy
          4
          5 NUM_BOLETO = 50
          6 TIEMPO_SIMULACION = 120
          7
          8 def ventaBoletos(env, num_boletos, pelicula, teatro):
          9     with teatro.contador.request() as turno:
         10         resultado = yield turno | teatro.sold_out[pelicula]
         11         if turno not in resultado:
         12             teatro.num_renegados[pelicula] += 1
         13         return
         14         if teatro.num_boletos[pelicula] < num_boletos:
         15             yield env.timeout(0.5)
         16         return
         17         teatro.num_boletos[pelicula] -= num_boletos
         18         if teatro.num_boletos[pelicula] < 2:
         19             teatro.sold_out[pelicula].succeed()
         20             teatro.tiempo_agotado[pelicula] = env.now
         21             teatro.num_boletos[pelicula] = 0
         22     yield env.timeout(1)
```

```

23 |
24 |
25 def llegadaClientes(env, teatro):
26     while True:
27         yield env.timeout(random.expovariate(1/0.5))
28         pelicula = random.choice(teatro.peliculas)
29         num_boletos = random.randint(1, 6)
30         if teatro.num_boletos[pelicula]:
31             env.process(ventaBoletos(env, num_boletos, pelicula, teat
32 |
33 |
34 Teatro = collections.namedtuple('Teatro', 'contador, peliculas,
35 |
36 print('Teatro Carlos Crespi - UPS')
37 env = simpy.Environment()
38 |
39 contador = simpy.Resource(env, capacity=1)
40 peliculas = ['Conjuro 3', 'Rapidos y Furiosos 10', 'Pulp Fiction
41 num_boletos = {pelicula: NUM_BOLETO for pelicula in peliculas}
42 sold_out = {pelicula: env.event() for pelicula in peliculas}
43 tiempo_agotado = {pelicula: None for pelicula in peliculas}
44 num_renegados = {pelicula: 0 for pelicula in peliculas}
45 |
46 teatro = Teatro(contador, peliculas, num_boletos, sold_out, tier
47 env.process(llegadaClientes(env, teatro))
48 env.run(until=TIEMPO_SIMULACION)
49 |
50 # Analisis y resultados
51 for pelicula in peliculas:
52     if teatro.sold_out[pelicula]:
53         print('Pelicula: %s se agoto en el tiempo %.1f despues de s
54         print('Numero de personas que salieron de la fila/rene
55

```

Teatro Carlos Crespi - UPS

Pelicula: Conjuro 3 se agoto en el tiempo 38.4 despues de salir a la venta

Numero de personas que salieron de la fila/renegados 14

Pelicula: Rapidos y Furiosos 10 se agoto en el tiempo 26.9 despues de salir a la venta

Numero de personas que salieron de la fila/renegados 8

Pelicula: Pulp Fictions se agoto en el tiempo 37.4 despues de salir a la venta

Numero de personas que salieron de la fila/renegados 18

Modificamos para agregar probabilidad a las peliculas al momento de comprar un boleto

```

In [41]: 1 import collections
          2 import random
          3 import simpy
          4 import matplotlib.pyplot as plt
          5 VELOCIDAD_LLEGADA = 42
          6 NUM_BOLETO = 50
          7 TIEMPO_SIMULACION = 120
          8
          9 veces_pelicula_vista = [0,0,0]
         10 def ventaBoletos(env, num_boletos, pelicula, teatro):
         11     with teatro.contador.request() as turno:
         12         resultado = yield turno | teatro.sold_out[pelicula]

```

```

13     if turno not in resultado:
14         teatro.num_renegados[pelicula] += 1
15         return
16     if teatro.num_boletos[pelicula] < num_boletos:
17         yield env.timeout(0.5)
18         return
19     teatro.num_boletos[pelicula] -= num_boletos
20     if teatro.num_boletos[pelicula] < 2:
21         teatro.sold_out[pelicula].succeed()
22         teatro.tiempo_agotado[pelicula] = env.now
23         teatro.num_boletos[pelicula] = 0
24     yield env.timeout(1)
25
26
27 def llegadaClientes(env, teatro):
28     while True:
29         yield env.timeout(random.expovariate(1/0.5))
30         pelicula = random.choices(teatro.peliculas, teatro.probabil
31         #SE OBTIENE DATOS PARA LA GRAFICA
32         if pelicula[0] == peliculas[0]:
33             veces_pelicula_vista[0] +=1
34         elif pelicula[0]== peliculas[1]:
35             veces_pelicula_vista[1] +=1
36         elif pelicula[0] == peliculas[2]:
37             veces_pelicula_vista[2] +=1
38
39         num_boletos = random.randint(1, 6)
40         if teatro.num_boletos[pelicula[0]]:
41             env.process(ventaBoletos(env, num_boletos, pelicula[0], t
42
43
44 Teatro = collections.namedtuple('Teatro', 'contador, peliculas,
45
46 print('Teatro Carlos Crespi - UPS')
47 env = simpy.Environment()
48 contador = simpy.Resource(env, capacity=1)
49 peliculas = ['Conjuro 3', 'Rapidos y Furiosos 10', 'Pulp Fiction
50 probabilidad=[0.2, 0.3, 0.1]
51 num_boletos = {pelicula: NUM_BOLETO for pelicula in peliculas}
52 sold_out = {pelicula: env.event() for pelicula in peliculas}
53 tiempo_agotado = {pelicula: None for pelicula in peliculas}
54 num_renegados = {pelicula: 0 for pelicula in peliculas}
55
56 teatro = Teatro(contador, peliculas, probabilidad, num_boletos,
57 env.process(llegadaClientes(env, teatro))
58 env.run(until=TIEMPO_SIMULACION)
59 # Analisis y resultados
60 for pelicula in peliculas:
61     if teatro.sold_out[pelicula]:
62         print('Pelicula: %s se agoto en el tiempo %.1f despues de s
63         print('Numero de personas que salieron de la fila/renegados
64
65
66 #SE GRAFICA EL HISTOGRAMA
67 plt.title('HISTOGRAMA DE VENTAS DE PELICULAS ')
68 plt.hist(veces_pelicula_vista, bins=40, width=1, alpha = 1, edgeco
69 plt.grid(axis='y', alpha = 0.75)
70 s = "Películas :", peliculas[2], "tiene > ", veces_pelicula_vista[
71 plt.xlabel(s, fontsize = 15)
72 plt.ylabel('FRECUENCIAS', fontsize = 15)

```

```
73 plt.show()
```

Teatro Carlos Crespi - UPS

Película: Conjuro 3 se agotó en el tiempo 39.7 después de salir a la venta

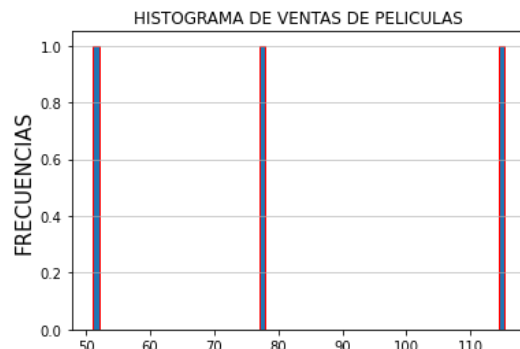
Número de personas que salieron de la fila/reneados 8

Película: Rápidos y Furiosos 10 se agotó en el tiempo 22.2 después de salir a la venta

Número de personas que salieron de la fila/reneados 3

Película: Pulp Fictions se agotó en el tiempo 50.1 después de salir a la venta

Número de personas que salieron de la fila/reneados 0



('Películas :', 'Pulp Fictions', 'tiene > ', 51, 'Conjuro 3', 'tiene > ', 78, 'Rápidos y Furiosos 10', 'tiene > ', 116)

```
In [2]: 1 #SE GRAFICA LA SIMULACION DE TEATRO
        2 import pygame, sys
        3 pygame.init()
        4
        5 #SE DEFINE LOS COLORES
        6 BLACK = (0,0,0)
        7 WHITE = (255,255,255)
        8 GREEN = (0,255,0)
        9 RED = (255,0,0)
       10 BLUE = (0,0,255)
       11 size=(800,500)
       12 screen = pygame.display.set_mode(size)
       13 screen.fill(WHITE)
       14
       15 for x in range(100,700,100):
       16     pygame.draw.rect(screen,BLACK,(x,230,50,50))
       17     pygame.draw.line(screen,GREEN,(x,0),(x,100),5)
       18 pygame.display.flip()
```

Tarea/Practica

Al modelo de simulación de la venta de boletos de un teatro/cine agregar la simulación gráfica 2D/3D utilizando cualquier librería gráfica, por ejemplo pygame/tkinter, etc. generar un informe en formato PDF que me permita evidenciar la simulación y generar una gráfica/reporte de los resultados.