

# Simulación en Python usando Simpy: Llegadas y servicio

*Realizaremos una simulación en Python usando Simpy de eventos discretos (simulación de llegadas y servicio).*

El ejemplo que vamos a utilizar es el siguiente: Una peluquería tiene un peluquero que se demora entre 15 y 30 minutos por corte. La peluquería recibe en promedio 3 clientes por hora (es decir, uno cada 20 minutos). Se desea simular las llegadas y servicios de 5 clientes.

Una vez realizado el código, podremos variar cualquiera de éstos parámetros (tener varios peluqueros, cambiar el tiempo que se demoran por corte, simular para n clientes o para un tiempo específico como una cierta cantidad de minutos, etc).

Antes de realizar la simulación en Python usando Simpy, vamos a hacer los cálculos manuales de acuerdo a las técnicas de Simulación. Esto nos permitirá tener un modelo matemático que nos confirme los resultados de nuestra simulación.

## Simulación de eventos discretos

El tiempo entre llegadas de los clientes a la peluquería es exponencial con media de 20 minutos.

La peluquería es atendida por un solo peluquero y se lleva entre 15 y 30 minutos, distribuidos de una manera uniforme, para realizar un corte de pelo.

Los clientes son atendidos con base en la disciplina FIFO (primero en llegar, primero en salir).

De modo que **las llegadas son calculadas de acuerdo a la siguiente fórmula:**

$$T\_LLEGADAS = -\lambda \ln(R)$$

Donde:

$\lambda$  es el promedio de llegadas, en nuestro ejemplo es de 20 minutos.

**Ln** es la operación de Logaritmo Natural

**R** es un número pseudoaleatorio

*Los tiempos de servicio son calculados de acuerdo a la siguiente fórmula:*

$$\text{tiempo\_corte} = \text{TIEMPO\_CORTE\_MIN} + (\text{TIEMPO\_CORTE\_MAX} - \text{TIEMPO\_CORTE\_MIN}) R$$

Esto es: el mínimo de tiempo que se demora el peluquero, en nuestro ejemplo es 15, más la diferencia entre el máximo y el mínimo, en nuestro ejemplo serían 15 minutos (30 menos 15), multiplicado por un número pseudoaleatorio. El resultado nos dará un número entre 15 y 30.

Los números pseudoaleatorios que utilizaremos serán los siguientes:

**0.5391, 0.2892, 0.6536, 0.2573, 0.6416, 0.0300, 0.2100, 0.3972, 0.9888, 0.4615**

Utilizaremos los primeros 5 para calcular las llegadas y los últimos 5 para calcular los tiempos de servicio.

## **Cálculo de los tiempos de llegada:**

$$\text{Cliente 1. } -20 * \ln(0.5391) = 12.36$$

$$\text{Cliente 2. } -20 * \ln(0.2892) = 24.81 \quad (\text{le sumamos 12.36 el resultado es: 37.17})$$

$$\text{Cliente 3. } -20 * \ln(0.6536) = 8.50 + 37.17 = 45.67$$

$$\text{Cliente 4. } -20 * \ln(0.2573) = 27.15 + 45.67 = 72.83$$

$$\text{Cliente 5. } -20 * \ln(0.6416) = 8.87 + 72.83 = 81.70$$

**Breve explicación:** Entre el minuto 0 y la llegada del cliente 1 pasaron 12.36 minutos. Entre el minuto 12.36 y la llegada del cliente 2 pasaron 24.81 minutos, por lo que el cliente 2 llegó en el minuto 37.17.

## Cálculo de los tiempos de servicio:

Cliente 1.  $15 + (15 * 0.0300) = 15.45$

Cliente 2.  $15 + (15 * 0.2100) = 18.15$

Cliente 3.  $15 + (15 * 0.3972) = 20.96$

Cliente 4.  $15 + (15 * 0.9888) = 29.83$

Cliente 5.  $15 + (15 * 0.4615) = 21.92$

**Breve explicación:** El cliente 1 demoró 15.45 minutos, el cliente 2 demoró 18.15 minutos. El cliente que más demoró fue el 4, demoró 29.83 minutos. Como puedes notar, el tiempo de servicio va de 15 a 30 minutos.

Con éstos datos podemos construir la siguiente tabla:

| Cliente | Llegada | Duración del servicio | Salida | Tiempo de espera |
|---------|---------|-----------------------|--------|------------------|
| 1       | 12.36   | 15.45                 | 27.81  | 0                |
| 2       | 37.17   | 18.15                 | 55.32  | 0                |
| 3       | 45.67   | 20.96                 | 76.28  | 9.65             |
| 4       | 72.83   | 29.83                 | 106.11 | 3.45             |
| 5       | 81.70   | 21.92                 | 128.03 | 24.41            |
|         |         | 106.31                |        | 37.51            |

Aplicando las fórmulas correspondientes podemos calcular la longitud promedio de la cola, el tiempo de espera promedio, y el uso promedio de la instalación.

**Longitud promedio de la cola** =  $37.51 / 128.03 = 0.29$

**Tiempo de espera promedio** =  $37.51 / 5 = 7.50$  minutos

**Uso promedio de la instalación** =  $106.31 / 128.03 = 0.83$  (La peluquería trabaja un 83% del tiempo total)

## Simulación en Python usando Simpy de una peluquería

Se considera que el lector ya ha utilizado Python, por lo que no incluyo la explicación de la programación en Python, y nos centraremos en las instrucciones propias de Simpy y la lógica del programa.

## Instalación de Simpy para Simulación en Python

Descarga Simpy desde su sitio web:

<https://pypi.python.org/pypi/simpy#downloads>

Una vez descargado, guarda el archivo en una ubicación accesible.

Abre esa ubicación (desde Windows escribe cd seguido de la ruta donde guardaste la carpeta de Simpy).

**\$ cd donde/colocaste/simpy/**

Ahora escribe el siguiente comando

**\$ python setup.py install**

## Programa de simulación en Python usando Simpy

### Primera parte: Bibliotecas a utilizar

```
import random
import math
```

```
import simpy
```

Usaremos random para generar números pseudoaleatorios, math para usar la función de logaritmo natural que utilizaremos para obtener las llegadas de los clientes a la peluquería, y simpy para tener un entorno de simulación.

## Segunda parte: Constantes y variables a utilizar

```
SEMILLA = 30

NUM_PELUQUEROS = 1

TIEMPO_CORTE_MIN = 15

TIEMPO_CORTE_MAX = 30

T_LLEGADAS = 20

TIEMPO_SIMULACION = 120

TOT_CLIENTES = 5


te = 0.0 # tiempo de espera total

dt = 0.0 # duracion de servicio total

fin = 0.0 # minuto en el que finaliza
```

El valor SEMILLA puede ser cualquiera, el que estamos utilizando nos proporciona los números pseudoaleatorios que utilizamos en la simulación manual.

La constante TIEMPO\_SIMULACION no la utilizaremos en este ejemplo, pero nos puede servir más adelante si, en vez de simular 5 clientes, queremos simular una cantidad de tiempo (120 minutos, por ejemplo).

## Tercera parte: Procedimientos

```
def cortar(cliente):

    global dt #Para poder acceder a la variable dt declarada anteriormente
```

```

R = random.random() # Obtiene un numero aleatorio y lo guarda en R

tiempo = TIEMPO_CORTE_MAX - TIEMPO_CORTE_MIN

tiempo_corte = TIEMPO_CORTE_MIN + (tiempo*R) # Distribucion uniforme

yield env.timeout(tiempo_corte) # deja correr el tiempo n minutos

print(" \o/ Corte listo a %s en %.2f minutos" % (cliente,tiempo_corte))

dt = dt + tiempo_corte # Acumula los tiempos de uso de la i

def cliente (env, name, personal ):

    global te

    global fin

    llega = env.now # Guarda el minuto de llegada del cliente

    print ("---> %s llego a peluqueria en minuto %.2f" % (name, llega))

    with personal.request() as request: # Espera su turno

        yield request # Obtiene turno

        pasa = env.now # Guarda el minuto cuando comienza a ser atendido

        espera = pasa - llega # Calcula el tiempo que espero

        te = te + espera # Acumula los tiempos de espera

        print ("**** %s pasa con peluquero en minuto %.2f habiendo esperado %.2f
" % (name, pasa, espera))

        yield env.process(cortar(name)) # Invoca al proceso cortar

        deja = env.now #Guarda el minuto en que termina el proceso cortar

        print ("<--- %s deja peluqueria en minuto %.2f" % (name, deja))

        fin = deja # Conserva globalmente el ultimo minuto de la simulación

def principal (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 u
no y otro

            i += 1

            env.process(cliente(env, 'Cliente %d' % i, personal))

```

El primer procedimiento invocado es el de principal, que a su vez manda a llamar a cliente, que a su vez manda a llamar a cortar.

El procedimiento cortar, calcula el tiempo que se demora con cada cliente. El procedimiento cliente muestra el minuto en que el cliente llega, espera a ser atendido, el minuto en que el cliente comienza a ser atendido, y el minuto en el que termina de ser atendido y deja la instalación.

El procedimiento principal hace un ciclo para n cantidad de clientes, en nuestro ejemplo, 5, y calcula las llegadas (que siguen una distribución exponencial).

## Cuarta parte: Programa principal

```

print ("----- Bienvenido Simulación Peluquería -----")

random.seed (SEMILLA) # Cualquier valor

env = simpy.Environment() # Crea el objeto entorno de simulación

personal = simpy.Resource(env, NUM_PELUQUEROS) #Crea los recursos (peluqueros)

env.process(principal(env, personal)) #Invoca el proceso principal

env.run() #Inicia la simulación

```

El programa inicia con un mensaje de Bienvenida, y crea un entorno de simulación. Crea los recursos, en este caso la cantidad de él o los peluqueros, y manda a llamar al proceso principal lo que da inicio a la simulación

## Quinta parte: Calculo de indicadores

```
print ("\n-----")

print ("\nIndicadores obtenidos: ")

lpc = te / fin

print ("\nLongitud promedio de la cola: %.2f" % lpc)

tep = te / TOT_CLIENTES

print ("Tiempo de espera promedio = %.2f" % tep)

upi = (dt / fin) / NUM_PELUQUEROS

print ("Uso promedio de la instalacion = %.2f" % upi)

print ("\n-----")
```

Calculamos los indicadores utilizando las fórmulas ya mencionadas.

## Resultado de la ejecución de la simulación en Python usando simpy

```
Terminal — zsh — 80x31

----- Bienvenido Simulacion Peluqueria -----
--> Cliente 1 llego a peluqueria en minuto 12.36
**** Cliente 1 pasa con peluquero en minuto 12.36 habiendo esperado 0.00
\o/ Corte listo a Cliente 1 en 15.45 minutos
<--- Cliente 1 deja peluqueria en minuto 27.81
--> Cliente 2 llego a peluqueria en minuto 37.17
**** Cliente 2 pasa con peluquero en minuto 37.17 habiendo esperado 0.00
--> Cliente 3 llego a peluqueria en minuto 45.67
\o/ Corte listo a Cliente 2 en 18.15 minutos
<--- Cliente 2 deja peluqueria en minuto 55.32
**** Cliente 3 pasa con peluquero en minuto 55.32 habiendo esperado 9.65
--> Cliente 4 llego a peluqueria en minuto 72.83
\o/ Corte listo a Cliente 3 en 20.96 minutos
<--- Cliente 3 deja peluqueria en minuto 76.28
**** Cliente 4 pasa con peluquero en minuto 76.28 habiendo esperado 3.45
--> Cliente 5 llego a peluqueria en minuto 81.70
\o/ Corte listo a Cliente 4 en 29.83 minutos
<--- Cliente 4 deja peluqueria en minuto 106.11
**** Cliente 5 pasa con peluquero en minuto 106.11 habiendo esperado 24.41
\o/ Corte listo a Cliente 5 en 21.92 minutos
<--- Cliente 5 deja peluqueria en minuto 128.03

-----

Indicadores obtenidos:

Longitud promedio de la cola: 0.29
Tiempo de espera promedio = 7.50
Uso promedio de la instalacion = 0.83

-----
```



Como podemos observar, debido a que se utilizaron los mismos números pseudoaleatorios obtenemos los mismos resultados. Podemos ahora variar a nuestro gusto los parámetros proporcionados al programa, por ejemplo, la cantidad de clientes, la cantidad de Peluqueros, o cuánto se demoran éstos en un corte de cabello.

## Otras instrucciones

Si deseamos que la simulación se ejecute durante un tiempo específico, en vez de hacerlo para  $n$  clientes:

En la línea **env.run** cambiamos la llamada a la siguiente orden:

```
env.run(until = TIEMPO_SIMULACION)
```

Donde **TIEMPO\_SIMULACION** es una variable declarada al principio del programa.

También debemos ajustar el procedimiento llamado principal. En vez de: **for i in range(TOT\_CLIENTES):**

Usaremos un ciclo infinito:

**While True:**

Si deseamos que el programa se ejecute en tiempo real, cambiemos la línea `env = simpy.Environment()` por

```
env = simpy.rt.RealtimeEnvironment(factor=1.0)
```

## Conclusión

La presente simulación en Python usando Simpy es una herramienta para obtener resultados en forma más sencilla que si lo hiciéramos manualmente. Es también muy flexible y se pueden crear programas más elaborados que si utilizáramos una hoja de cálculo.

Python es un lenguaje multiplataforma, además, es software libre, por lo que es ideal para cualquier proyecto académico.

Es posible adaptar este programa para resolver diferentes problemas y se puede ampliar para obtener más datos cuya solución manual o matemática pudiera ser complicada.

## EJERCICIO

Se implementa un proceso SimPy en la clase de estacionamiento para modelar las llegadas de automóviles. Este proceso continúa durante un número definido de llegadas de automóviles. Cada automóvil llega después de un tiempo entre llegadas distribuido aleatoriamente. Luego ocupa un espacio disponible durante un período de tiempo distribuido aleatoriamente. Después de eso, se libera la plaza de aparcamiento.

Antes de realizar la simulación en Python usando Simpy, deberá hacer los cálculos manuales de acuerdo a las técnicas de Simulación. Esto nos permitirá tener un modelo matemático que nos confirme los resultados de nuestra simulación.

Al ejecutar este modelo para 10 llegadas de automóviles, para un estacionamiento con 10 espacios de estacionamiento, se genera un resultado como el siguiente:

El auto 2 llega a las 4

Coche 2 aparcamientos a las 4

El auto 1 llega a 2

El coche 1 se aparca en 2

El coche 1 sale a las 3

El coche 3 llega a las 7

Coche 3 aparcamientos a las 7

El auto 4 llega a las 8

Coche 4 aparca a las 8

El coche 2 sale a las 9

El coche 5 llega a las 9

Coche 5 aparca a las 9

El coche 4 sale a las 10

El auto 6 llega a las 11

Coche 6 aparca a las 11

El coche 3 sale a las 12

El auto 7 llega a las 12

Coche 7 aparca a las 12

El coche 5 sale a las 13

El coche 6 sale a las 13

El coche 8 llega a las 16.

Coche 8 aparca a las 16

El coche 7 sale a las 17

El coche 8 sale a las 17

El coche 9 llega a las 20

Coche 9 aparca a las 20

El coche 10 llega a las 22.

Coche 10 aparca a las 22

El coche 10 sale a las 24

El coche 9 sale a las 25