

# Deber Simpy

## 1. Instalación de Simpy

Simpy en python pip3 install -U simpy

```
In [2]: import simpy
def reloj(env, nombre, tiempo):
    while True:
        print(nombre, ">", env.now)
        yield env.timeout(tiempo)

env=simpy.Environment()
env.process(reloj(env, 'R. Rapido',0.25))
env.process(reloj(env, 'R. Lento',0.5))

env.run(until=2.1)
```

```
R. Rapido > 0
R. Lento > 0
R. Rapido > 0.25
R. Lento > 0.5
R. Rapido > 0.5
R. Rapido > 0.75
R. Lento > 1.0
R. Rapido > 1.0
R. Rapido > 1.25
R. Lento > 1.5
R. Rapido > 1.5
R. Rapido > 1.75
R. Lento > 2.0
R. Rapido > 2.0
```

## Ejemplo del negocio de lavado de autos

Este ejemplo permite simular un negocio de lavado de automóviles. De igual forma, es importante destacar que este ejemplo aborda los siguientes puntos:

## Características del sistema real a simular

SimPy es un marco de simulación de eventos discretos basado en procesos basado en Python estándar. Su despachador de eventos se basa en los generadores de Python y también se puede utilizar para redes asíncronas o para implementar sistemas multiagente (con comunicación tanto simulada como real).

Los procesos en SimPy son funciones simples del generador de Python y se utilizan para modelar componentes activos como clientes, vehículos o agentes. SimPy también proporciona varios tipos de recursos compartidos para modelar puntos de congestión de capacidad limitada (como servidores, contadores de pago y túneles). A partir de la versión 3.1, también proporcionará capacidades de supervisión para ayudar a recopilar estadísticas sobre recursos y procesos.

Las simulaciones se pueden realizar "lo más rápido posible", en tiempo real (tiempo de reloj de pared) o pasando manualmente por los eventos.

Aunque teóricamente es posible hacer simulaciones continuas con SimPy, no tiene características para realizar eso. Sin embargo, SimPy es excesivo para las simulaciones con un tamaño de paso fijo en el que sus procesos no interactúan entre sí o con recursos compartidos; utilice un whileciclo simple en este caso.

La distribución SimPy contiene tutoriales, documentación detallada y una gran cantidad de ejemplos.

SimPy se lanza como software de código abierto bajo la licencia MIT . La primera versión se lanzó en diciembre de 2002.

```

In [3]: import simpy
import random

MAX_VEHICULOS = 57
NUM_MAQUINAS = 3
TIEMPO_LAVADO = 7
INTERVALO_LLEGADA = 9
TIEMPO_SIMULACION = 23

class Lavanderia(object):

    def __init__(self, environment, num_maquinas, tiempo_lavado):
        self.env=environment
        self.maquinas = simpy.Resource(environment, num_maquinas)
        self.tiempo_lavado = tiempo_lavado

    def lavar_vehiculo(self, vehiculo):
        yield self.env.timeout(TIEMPO_LAVADO)
        print('Removido {0d%} suciedad vehiculo => %s ' \
              % (random.randint(30,90), vehiculo))

def llegada_vehiculo(env, nombre, lavanderia):
    print('Llega vehiculo: %s a la hora %.2f.' % (nombre, env.now))
    with lavanderia.maquinas.request() as maquina:
        yield maquina
        print('Entra vehiculo a lavarse: %s a la hora %.2f.' % (nombre, env.now))
        yield env.process(lavanderia.lavar_vehiculo(nombre))
        print('Vehiculo [%s] lavado a las %.2f.' % (nombre, env.now))

def ejecutar_simulacion(env, num_maquinas, tiempo_lavado, intervalo):
    lavanderia=Lavanderia(env, num_maquinas, tiempo_lavado)
    for i in range(5):
        env.process(llegada_vehiculo(env, 'Vehiculo-%d'%(i+1),lavanderia))
    while True:
        yield env.timeout(random.randint(intervalo-2, intervalo+2))
        i+=1
        env.process(llegada_vehiculo(env, 'Vehiculo-%d'%(i+1),lavanderia))

print('Lavanderia UPS')
env=simpy.Environment()
env.process(ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO, INTERVALO_LLEGADA))
env.run(until = TIEMPO_SIMULACION)

```

```

Lavanderia UPS
Llega vehiculo: Vehiculo-1 a la hora 0.00.
Llega vehiculo: Vehiculo-2 a la hora 0.00.
Llega vehiculo: Vehiculo-3 a la hora 0.00.
Llega vehiculo: Vehiculo-4 a la hora 0.00.
Llega vehiculo: Vehiculo-5 a la hora 0.00.
Entra vehiculo a lavarse: Vehiculo-1 a la hora 0.00.
Entra vehiculo a lavarse: Vehiculo-2 a la hora 0.00.
Entra vehiculo a lavarse: Vehiculo-3 a la hora 0.00.
Llega vehiculo: Vehiculo-6 a la hora 7.00.
Removido {70%} suciedad vehiculo => Vehiculo-1
Removido {51%} suciedad vehiculo => Vehiculo-2
Removido {31%} suciedad vehiculo => Vehiculo-3
Vehiculo [Vehiculo-1] lavado a las 7.00.
Vehiculo [Vehiculo-2] lavado a las 7.00.
Vehiculo [Vehiculo-3] lavado a las 7.00.
Entra vehiculo a lavarse: Vehiculo-4 a la hora 7.00.

```

Entra vehiculo a lavarse: Vehiculo-5 a la hora 7.00.  
Entra vehiculo a lavarse: Vehiculo-6 a la hora 7.00.  
Removido {56%} suciedad vehiculo => Vehiculo-4  
Removido {47%} suciedad vehiculo => Vehiculo-5  
Removido {81%} suciedad vehiculo => Vehiculo-6  
Vehiculo [Vehiculo-4] lavado a las 14.00.  
Vehiculo [Vehiculo-5] lavado a las 14.00.  
Vehiculo [Vehiculo-6] lavado a las 14.00.  
Llega vehiculo: Vehiculo-7 a la hora 16.00.  
Entra vehiculo a lavarse: Vehiculo-7 a la hora 16.00.