

Practica-Simpy

June 5, 2021

1 Tarea en clases

1.0.1 2.3. Visualizando la información de la simulación

A continuación modificaremos el ejemplo propuesto en el libro guía de SimPy a fin de incluir una gráfica que nos provea información sobre la simulación realizada

1.1 Práctica 1 - SimPy SED

- Considerar 3 tipos de vehículos con 3 diferentes tiempos de lavado:
 - Bus: 20 minutos
 - Automóvil: 7 minutos
 - Furgoneta: 12 minutos
- Considere que a partir de la entrada del vehículo para ser lavado, deberán tomarse en cuenta los siguientes tiempos correspondientes al proceso de traslado desde la recepción a la máquina de lavado:
 - Bus: 4 minutos
 - Automóvil: 1 minuto
 - Furgoneta: 2 minutos
- Realice una gráfica comparativa de la simulación con 3 y con 4 máquinas de lavado

2 3 maquinas

```
[17]: import simpy
import random
import matplotlib.pyplot as pp

# Maximo de vehiculos que puede recibir el negocio
MAX_VEHICULOS = 57
# Total de maquinas de lavado con que cuenta el negocio
NUM_MAQUINAS = 3
# Tiempo que tarda en trasladarse un vehiculo (minutos)
TRASLADO_BUS = 4
TRASLADO_AUTOMOVIL= 1
TRASLADO_FURGONETA= 2
# Tiempo que tarda en lavarse un vehiculo (minutos)
TIEMPO_BUS = 20
TIEMPO_AUTOMOVIL= 7
```

```

TIEMPO_FURGONETA= 12
# Intervalo de tiempo en que llegan vehiculos (minutos)
INTERVALO_LLEGADA = 9
# Tiempo de simulación
TIEMPO_SIMULACION = 27

# Creamos un diccionario para almacenar las horas en que se lavan los vehiculos
tiempo={}

class Lavanderia(object):

    def __init__(self, environment, num_maquinas,tiempo_bus,tiempo_automovil,
        tiempo_furgoneta,traslado_bus,traslado_automovil,traslado_furgoneta):
        # Guardamos como variable el entorno de ejecucion
        self.env=environment
        # Creamos el recurso que representa las maquinas
        self.maquinas = simpy.Resource(environment, num_maquinas)
        # Variable para el tiempo de lavado
        self.tiempo_bus = tiempo_bus
        self.tiempo_automovil = tiempo_automovil
        self.tiempo_furgoneta = tiempo_furgoneta
        self.traslado_bus = traslado_bus
        self.traslado_automovil = traslado_automovil
        self.traslado_furgoneta = traslado_furgoneta

    def lavar_vehiculo(self, vehiculo):

        automotor=vehiculo.split('-')

        if(automotor[0] == 'Bus'):
            yield self.env.timeout(TIEMPO_BUS)

        if(automotor[0] == 'Automovil'):
            yield self.env.timeout(TIEMPO_AUTOMOVIL)

        if(automotor[0] == 'Furgoneta'):
            yield self.env.timeout(TIEMPO_FURGONETA)

        k=env.now
        if k in tiempo:
            tiempo[k]=tiempo[k]+1
        else:
            tiempo[k]=1

        # Simulamos que se ha limpiado parte (%) de la suciedad del vehiculo
        # Para el % generamos un entero entre 30 y 90

```

```

print('Removido  {%d%%} suciedad vehiculo => %s ' \
      % (random.randint(30,90), vehiculo))

def trasladar_vehiculo(self, vehiculo):

    automotor=vehiculo.split('-')

    if(automotor[0] == 'Bus'):
        yield self.env.timeout(TRASLADO_BUS)

    if(automotor[0] == 'Automovil'):
        yield self.env.timeout(TRASLADO_AUTOMOVIL)

    if(automotor[0] == 'Furgoneta'):
        yield self.env.timeout(TRASLADO_FURGONETA)

    k=env.now
    if k in tiempo:
        tiempo[k]=tiempo[k]+1
    else:
        tiempo[k]=1

def llegada_vehiculo(env, nombre, lavanderia):
    # 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 vehiculo: %s a la hora %.2f.' % (nombre, env.now))

    # Especificamos que vamos a usar un recurso (Resource) que representa
    # la maquina de lavado
    with lavanderia.maquinas.request() as maquina:
        # Ocupamos la maquina de lavado
        yield maquina
        #Indicamos que el vehiculo se esta trasladando para ser lavado
        print('Se traslada vehiculo: %s a la hora %.2f.' % (nombre, env.now))
        #Procesamos el traslado del vehiculo
        yield env.process(lavanderia.trasladar_vehiculo(nombre))
        # Indicamos que vehiculo entra a la lavanderia
        print('Entra vehiculo a lavarse: %s a la hora %.2f.' % (nombre, env.
→now))
        # Procesamos la operacion de lavado
        yield env.process(lavanderia.lavar_vehiculo(nombre))
        # Una vez que termina la llamada con 'yield', se indica que se ha
→lavado el vehiculo
        print('Vehiculo [%s] lavado a las %.2f.' % (nombre, env.now))

def ejecutar_simulacion(env, num_maquinas, tiempo_bus, tiempo_automovil,
                        tiempo_furgoneta, traslado_bus, traslado_automovil,

```

```

        traslado_furgoneta, intervalo):
lavanderia=Lavanderia(env, num_maquinas, tiempo_bus,tiempo_automovil,
                        tiempo_furgoneta,traslado_bus,traslado_automovil,
                        traslado_furgoneta)
# Creamos 5 llegadas de vehiculos iniciales
for i in range(5):
    env.process(llegada_vehiculo(env, 'Bus-%d'%(i+1),lavanderia))
    env.process(llegada_vehiculo(env, 'Automovil-%d'%(i+1),lavanderia))
    env.process(llegada_vehiculo(env, 'Furgoneta-%d'%(i+1),lavanderia))

# Ejecutamos la simulacion
while True:
    yield env.timeout(random.randint(intervalo-3, intervalo+3))
    i+=1
    # Mientras se lavan los vehiculos generamos mas vehiculos
    env.process(llegada_vehiculo(env, 'Bus-%d'%(i+1),lavanderia))
    env.process(llegada_vehiculo(env, 'Automovil-%d'%(i+1),lavanderia))
    env.process(llegada_vehiculo(env, 'Furgoneta-%d'%(i+1),lavanderia))

print('Lavanderia UPS')
# Inicializamos la semilla aleatoria
random.seed(77)

# Creamos el entorno de simulacion
env=simpy.Environment()
env.process(ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_BUS,TIEMPO_AUTOMOVIL,
                                ↵
                                ↪TIEMPO_FURGONETA,TRASLADO_BUS,TRASLADO_AUTOMOVIL,
                                TRASLADO_FURGONETA, INTERVALO_LLEGADA))

# Ejecutamos el proceso durante el tiempo de simulacion
env.run(until = TIEMPO_SIMULACION)

# Generamos la grafica
datos=sorted(tiempo.items())
x1, y1 =zip(*datos)
pp.plot(x1,y1,linewidth=2,color='red')
pp.scatter(x1,y1,color='blue')
pp.grid(True)
#pp.show()

pp.title('Grafica Final 3 maquinas')
pp.savefig('maquina3.png')
pp.show()

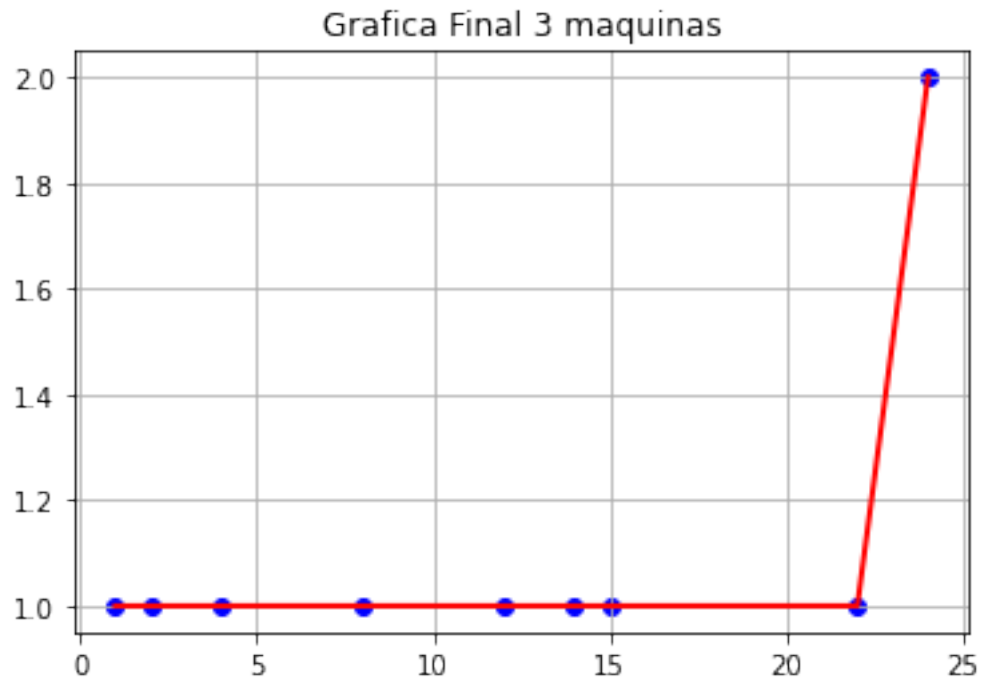
```

Lavanderia UPS

Llega vehiculo: Bus-1 a la hora 0.00.

Llega vehiculo: Automovil-1 a la hora 0.00.

Llega vehiculo: Furgoneta-1 a la hora 0.00.
Llega vehiculo: Bus-2 a la hora 0.00.
Llega vehiculo: Automovil-2 a la hora 0.00.
Llega vehiculo: Furgoneta-2 a la hora 0.00.
Llega vehiculo: Bus-3 a la hora 0.00.
Llega vehiculo: Automovil-3 a la hora 0.00.
Llega vehiculo: Furgoneta-3 a la hora 0.00.
Llega vehiculo: Bus-4 a la hora 0.00.
Llega vehiculo: Automovil-4 a la hora 0.00.
Llega vehiculo: Furgoneta-4 a la hora 0.00.
Llega vehiculo: Bus-5 a la hora 0.00.
Llega vehiculo: Automovil-5 a la hora 0.00.
Llega vehiculo: Furgoneta-5 a la hora 0.00.
Se traslada vehiculo: Bus-1 a la hora 0.00.
Se traslada vehiculo: Automovil-1 a la hora 0.00.
Se traslada vehiculo: Furgoneta-1 a la hora 0.00.
Entra vehiculo a lavarse: Automovil-1 a la hora 1.00.
Entra vehiculo a lavarse: Furgoneta-1 a la hora 2.00.
Entra vehiculo a lavarse: Bus-1 a la hora 4.00.
Removido {46%} suciedad vehiculo => Automovil-1
Vehiculo [Automovil-1] lavado a las 8.00.
Se traslada vehiculo: Bus-2 a la hora 8.00.
Llega vehiculo: Bus-6 a la hora 12.00.
Llega vehiculo: Automovil-6 a la hora 12.00.
Llega vehiculo: Furgoneta-6 a la hora 12.00.
Entra vehiculo a lavarse: Bus-2 a la hora 12.00.
Removido {42%} suciedad vehiculo => Furgoneta-1
Vehiculo [Furgoneta-1] lavado a las 14.00.
Se traslada vehiculo: Automovil-2 a la hora 14.00.
Entra vehiculo a lavarse: Automovil-2 a la hora 15.00.
Llega vehiculo: Bus-7 a la hora 20.00.
Llega vehiculo: Automovil-7 a la hora 20.00.
Llega vehiculo: Furgoneta-7 a la hora 20.00.
Removido {42%} suciedad vehiculo => Automovil-2
Vehiculo [Automovil-2] lavado a las 22.00.
Se traslada vehiculo: Furgoneta-2 a la hora 22.00.
Removido {82%} suciedad vehiculo => Bus-1
Vehiculo [Bus-1] lavado a las 24.00.
Entra vehiculo a lavarse: Furgoneta-2 a la hora 24.00.
Se traslada vehiculo: Bus-3 a la hora 24.00.



3 4 maquinas

```
[18]: import simpy
import random
import matplotlib.pyplot as pp

# Maximo de vehiculos que puede recibir el negocio
MAX_VEHICULOS = 57
# Total de maquinas de lavado con que cuenta el negocio
NUM_MAQUINAS = 4
# Tiempo que tarda en trasladarse un vehiculo (minutos)
TRASLADO_BUS = 4
TRASLADO_AUTOMOVIL= 1
TRASLADO_FURGONETA= 2
# Tiempo que tarda en lavarse un vehiculo (minutos)
TIEMPO_BUS = 20
TIEMPO_AUTOMOVIL= 7
TIEMPO_FURGONETA= 12
# Intervalo de tiempo en que llegan vehiculos (minutos)
INTERVALO_LLEGADA = 9
# Tiempo de simulación
TIEMPO_SIMULACION = 27
```

```

# Creamos un diccionario para almacenar las horas en que se lavan los vehiculos
tiempo={}

class Lavanderia(object):

    def __init__(self, environment, num_maquinas, tiempo_bus, tiempo_automovil,
        tiempo_furgoneta, traslado_bus, traslado_automovil, traslado_furgoneta):
        # Guardamos como variable el entorno de ejecucion
        self.env=environment
        # Creamos el recurso que representa las maquinas
        self.maquinas = simpy.Resource(environment, num_maquinas)
        # Variable para el tiempo de lavado
        self.tiempo_bus = tiempo_bus
        self.tiempo_automovil = tiempo_automovil
        self.tiempo_furgoneta = tiempo_furgoneta
        self.traslado_bus = traslado_bus
        self.traslado_automovil = traslado_automovil
        self.traslado_furgoneta = traslado_furgoneta

    def lavar_vehiculo(self, vehiculo):

        automotor=vehiculo.split('-')

        if(automotor[0] == 'Bus'):
            yield self.env.timeout(TIEMPO_BUS)

        if(automotor[0] == 'Automovil'):
            yield self.env.timeout(TIEMPO_AUTOMOVIL)

        if(automotor[0] == 'Furgoneta'):
            yield self.env.timeout(TIEMPO_FURGONETA)

        k=env.now
        if k in tiempo:
            tiempo[k]=tiempo[k]+1
        else:
            tiempo[k]=1

        # Simulamos que se ha limpiado parte (%) de la suciedad del vehiculo
        # Para el % generamos un entero entre 30 y 90
        print('Removido {%d%%} suciedad vehiculo => %s ' \
            % (random.randint(30,90), vehiculo))

    def trasladar_vehiculo(self, vehiculo):

        automotor=vehiculo.split('-')

```

```

    if(automotor[0] == 'Bus'):
        yield self.env.timeout(TRASLADO_BUS)

    if(automotor[0] == 'Automovil'):
        yield self.env.timeout(TRASLADO_AUTOMOVIL)

    if(automotor[0] == 'Furgoneta'):
        yield self.env.timeout(TRASLADO_FURGONETA)

    k=env.now
    if k in tiempo:
        tiempo[k]=tiempo[k]+1
    else:
        tiempo[k]=1

def llegada_vehiculo(env, nombre, lavanderia):
    # 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 vehiculo: %s a la hora %.2f.' % (nombre, env.now))

    # Especificamos que vamos a usar un recurso (Resource) que representa
    # la maquina de lavado
    with lavanderia.maquinas.request() as maquina:
        # Ocupamos la maquina de lavado
        yield maquina
        #Indicamos que el vehiculo se esta trasladando para ser lavado
        print('Se traslada vehiculo: %s a la hora %.2f.' % (nombre, env.now))
        #Procesamos el traslado del vehiculo
        yield env.process(lavanderia.trasladar_vehiculo(nombre))
        # Indicamos que vehiculo entra a la lavanderia
        print('Entra vehiculo a lavarse: %s a la hora %.2f.' % (nombre, env.
↪now))
        # Procesamos la operacion de lavado
        yield env.process(lavanderia.lavar_vehiculo(nombre))
        # Una vez que termina la llamada con 'yield', se indica que se ha
↪lavado el vehiculo
        print('Vehiculo [%s] lavado a las %.2f.' % (nombre, env.now))

def ejecutar_simulacion(env, num_maquinas, tiempo_bus, tiempo_automovil,
    tiempo_furgoneta, traslado_bus, traslado_automovil,
    traslado_furgoneta, intervalo):
    lavanderia=Lavanderia(env, num_maquinas, tiempo_bus, tiempo_automovil,
        tiempo_furgoneta, traslado_bus, traslado_automovil,
        traslado_furgoneta)
    # Creamos 5 llegadas de vehiculos iniciales
    for i in range(5):

```



```

env.process(llegada_vehiculo(env, 'Bus-%d'%(i+1),lavanderia))
env.process(llegada_vehiculo(env, 'Automovil-%d'%(i+1),lavanderia))
env.process(llegada_vehiculo(env, 'Furgoneta-%d'%(i+1),lavanderia))

# Ejecutamos la simulacion
while True:
    yield env.timeout(random.randint(intervalo-3, intervalo+3))
    i+=1
    # Mientras se lavan los vehiculos generamos mas vehiculos
    env.process(llegada_vehiculo(env, 'Bus-%d'%(i+1),lavanderia))
    env.process(llegada_vehiculo(env, 'Automovil-%d'%(i+1),lavanderia))
    env.process(llegada_vehiculo(env, 'Furgoneta-%d'%(i+1),lavanderia))

print('Lavanderia UPS')
# Inicializamos la semilla aleatoria
random.seed(77)

# Creamos el entorno de simulacion
env=simpy.Environment()
env.process(ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_BUS,TIEMPO_AUTOMOVIL,
                                ↵
                                ↪TIEMPO_FURGONETA,TRASLADO_BUS,TRASLADO_AUTOMOVIL,
                                TRASLADO_FURGONETA, INTERVALO_LLEGADA))

# Ejecutamos el proceso durante el tiempo de simulacion
env.run(until = TIEMPO_SIMULACION)

# Generamos la grafica
datos=sorted(tiempo.items())
x, y =zip(*datos)
pp.plot(x,y,linewidth=2,color='red')
pp.scatter(x,y,color='blue')
pp.grid(True)
#pp.show()

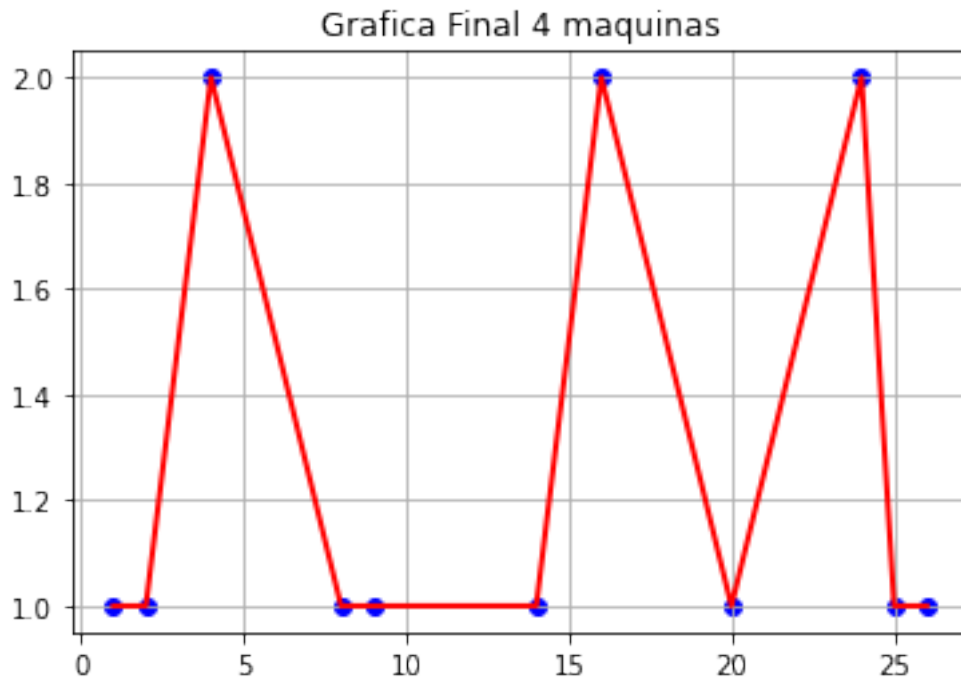
pp.title('Grafica Final 4 maquinas')
pp.savefig('maquina4.png')
pp.show()

```

Lavanderia UPS

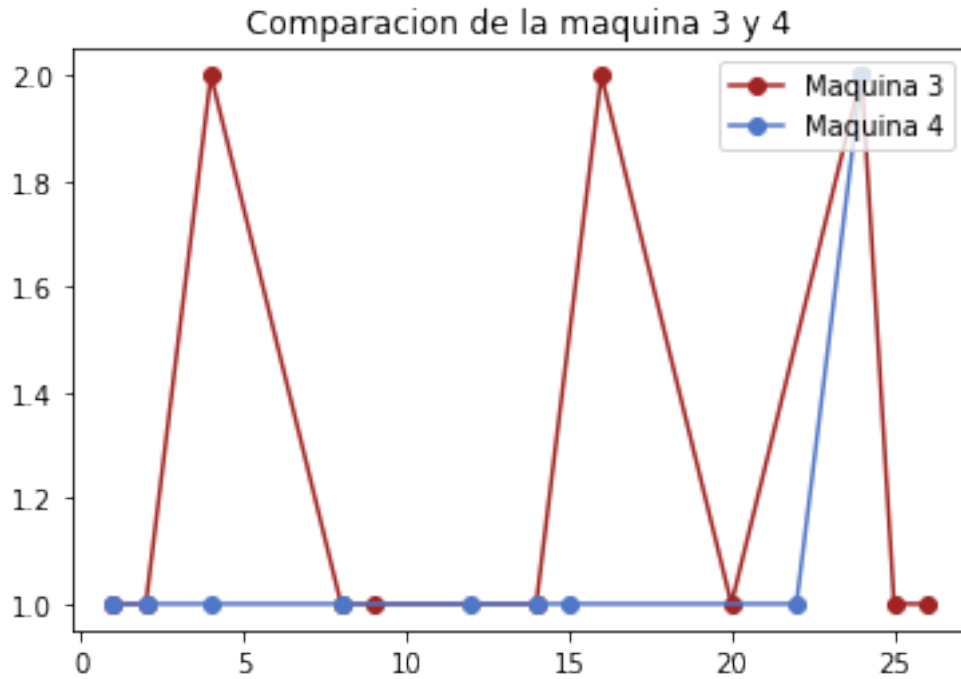
Llega vehiculo: Bus-1 a la hora 0.00.
Llega vehiculo: Automovil-1 a la hora 0.00.
Llega vehiculo: Furgoneta-1 a la hora 0.00.
Llega vehiculo: Bus-2 a la hora 0.00.
Llega vehiculo: Automovil-2 a la hora 0.00.
Llega vehiculo: Furgoneta-2 a la hora 0.00.
Llega vehiculo: Bus-3 a la hora 0.00.
Llega vehiculo: Automovil-3 a la hora 0.00.

Llega vehiculo: Furgoneta-3 a la hora 0.00.
Llega vehiculo: Bus-4 a la hora 0.00.
Llega vehiculo: Automovil-4 a la hora 0.00.
Llega vehiculo: Furgoneta-4 a la hora 0.00.
Llega vehiculo: Bus-5 a la hora 0.00.
Llega vehiculo: Automovil-5 a la hora 0.00.
Llega vehiculo: Furgoneta-5 a la hora 0.00.
Se traslada vehiculo: Bus-1 a la hora 0.00.
Se traslada vehiculo: Automovil-1 a la hora 0.00.
Se traslada vehiculo: Furgoneta-1 a la hora 0.00.
Se traslada vehiculo: Bus-2 a la hora 0.00.
Entra vehiculo a lavarse: Automovil-1 a la hora 1.00.
Entra vehiculo a lavarse: Furgoneta-1 a la hora 2.00.
Entra vehiculo a lavarse: Bus-1 a la hora 4.00.
Entra vehiculo a lavarse: Bus-2 a la hora 4.00.
Removido {46%} suciedad vehiculo => Automovil-1
Vehiculo [Automovil-1] lavado a las 8.00.
Se traslada vehiculo: Automovil-2 a la hora 8.00.
Entra vehiculo a lavarse: Automovil-2 a la hora 9.00.
Llega vehiculo: Bus-6 a la hora 12.00.
Llega vehiculo: Automovil-6 a la hora 12.00.
Llega vehiculo: Furgoneta-6 a la hora 12.00.
Removido {42%} suciedad vehiculo => Furgoneta-1
Vehiculo [Furgoneta-1] lavado a las 14.00.
Se traslada vehiculo: Furgoneta-2 a la hora 14.00.
Removido {45%} suciedad vehiculo => Automovil-2
Vehiculo [Automovil-2] lavado a las 16.00.
Entra vehiculo a lavarse: Furgoneta-2 a la hora 16.00.
Se traslada vehiculo: Bus-3 a la hora 16.00.
Llega vehiculo: Bus-7 a la hora 20.00.
Llega vehiculo: Automovil-7 a la hora 20.00.
Llega vehiculo: Furgoneta-7 a la hora 20.00.
Entra vehiculo a lavarse: Bus-3 a la hora 20.00.
Removido {82%} suciedad vehiculo => Bus-1
Removido {90%} suciedad vehiculo => Bus-2
Vehiculo [Bus-1] lavado a las 24.00.
Vehiculo [Bus-2] lavado a las 24.00.
Se traslada vehiculo: Automovil-3 a la hora 24.00.
Se traslada vehiculo: Furgoneta-3 a la hora 24.00.
Entra vehiculo a lavarse: Automovil-3 a la hora 25.00.
Entra vehiculo a lavarse: Furgoneta-3 a la hora 26.00.



```
[26]: pp.title('Comparacion de la maquina 3 y 4')  
  
pp.plot(x, y, color='#a12424', linestyle='--', marker='o')  
pp.plot(x1, y1, color='#5077C7', linestyle='--', marker='o')  
pp.legend(['Maquina 3', 'Maquina 4'], loc=1)
```

```
[26]: <matplotlib.legend.Legend at 0x2320726d490>
```



3.1 Referencias

- [1] Matloff, N. (2008). Introduction to Discrete-Event Simulation and the SimPy Language.
- [2] Team Simpy (2017). SimPy Documentation, Release 3.0.10, 2017. URL: <https://media.readthedocs.org/pdf/simpy/latest/simpy.pdf>