

Universidad Politecnica Salesiana

Nombre: Jessica Ñauta

Asignatura: Simulación

Entornos de soporte al desarrollo de simulaciones: Simpy - Parte 2

Generacion de graficar de atencion.

En base a la simulacion anterior generar una grafica de los autos atendidos.

```
In [6]: import simpy
import random
import matplotlib.pyplot as pp

%matplotlib inline

# 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 lavarse un vehiculo (minutos)
TIEMPO_LAVADO = 10
# Intervalo de tiempo en que llegan vehiculos (minutos)
INTERVALO_LLEGADA = 3
# 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_lavado):
        # 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_lavado = tiempo_lavado

    def lavar_vehiculo(self, vehiculo):
        # Este metodo representa el proceso de lavado del vehiculo.
        # Se ingresa el vehiculo y se lava

        # Simulamos el tiempo que tarda en lavarse el vehiculo
        # Es importante notar que la instruccion "yield" es distinta de "sleep"
        # ya que esta ultima bloquea el hilo de ejecucion durante 't' unidades de ti
        # mientras que 'yield' no bloquea el hilo de ejecucion, solo lo suspende mie
        # el evento de 'lavado' se realice
        yield self.env.timeout(random.randint(TIEMPO_LAVADO-5, TIEMPO_LAVADO+5))

        # 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),
```

```

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 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
        print('Vehiculo [%s] lavado a las %.2f.' % (nombre, env.now))

    tiempo[nombre]=env.now

def ejecutar_simulacion(env, num_maquinas, tiempo_lavado, intervalo):
    lavanderia=Lavanderia(env, num_maquinas, tiempo_lavado)
    # Creamos 5 llegadas de vehiculos iniciales
    for i in range(5):
        env.process(llegada_vehiculo(env, 'Vehiculo-%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, 'Vehiculo-%d'%(i+1),lavanderia))

print('Lavanderia UPS')

# Creamos el entorno de simulacion
env=sympy.Environment()
env.process(ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO, INTERVALO_LLEGADA))

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

print("Diccionario tiempo :")
print(tiempo)

```

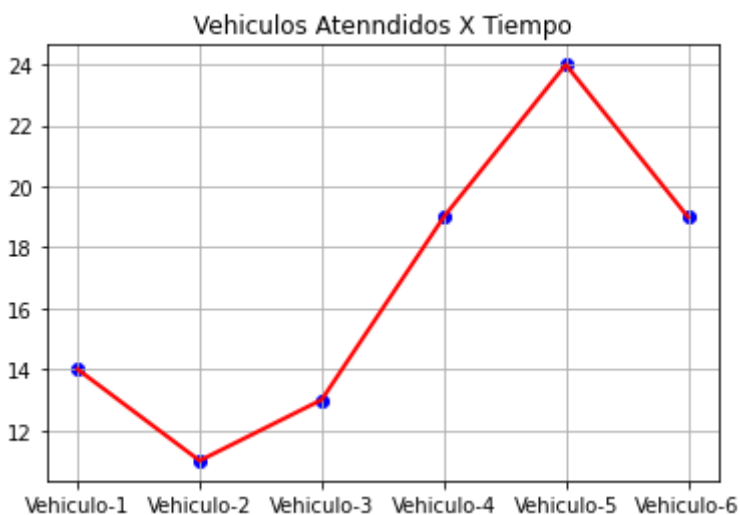
```

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 6.00.
Llega vehiculo: Vehiculo-7 a la hora 10.00.
Llega vehiculo: Vehiculo-8 a la hora 10.00.
Removido {47%} suciedad vehiculo => Vehiculo-2
Vehiculo [Vehiculo-2] lavado a las 11.00.
Entra vehiculo a lavarse: Vehiculo-4 a la hora 11.00.
Llega vehiculo: Vehiculo-9 a la hora 12.00.
Removido {66%} suciedad vehiculo => Vehiculo-3

```

Vehiculo [Vehiculo-3] lavado a las 13.00.
 Entra vehiculo a lavarse: Vehiculo-5 a la hora 13.00.
 Removido {66%} suciedad vehiculo => Vehiculo-1
 Vehiculo [Vehiculo-1] lavado a las 14.00.
 Entra vehiculo a lavarse: Vehiculo-6 a la hora 14.00.
 Llega vehiculo: Vehiculo-10 a la hora 18.00.
 Removido {85%} suciedad vehiculo => Vehiculo-4
 Removido {83%} suciedad vehiculo => Vehiculo-6
 Vehiculo [Vehiculo-4] lavado a las 19.00.
 Vehiculo [Vehiculo-6] lavado a las 19.00.
 Entra vehiculo a lavarse: Vehiculo-7 a la hora 19.00.
 Entra vehiculo a lavarse: Vehiculo-8 a la hora 19.00.
 Removido {42%} suciedad vehiculo => Vehiculo-5
 Llega vehiculo: Vehiculo-11 a la hora 24.00.
 Vehiculo [Vehiculo-5] lavado a las 24.00.
 Entra vehiculo a lavarse: Vehiculo-9 a la hora 24.00.
 Diccionario tiempo :
 {'Vehiculo-2': 11, 'Vehiculo-3': 13, 'Vehiculo-1': 14, 'Vehiculo-4': 19, 'Vehiculo-6': 19, 'Vehiculo-5': 24}

```
In [7]: # Generamos La grafica
datos=sorted(tiempo.items()) # Ordenamos Los datos
x, y =zip(*datos) # Obtener x(tiempo - clave) y el y(Numero de vehiculos atendidos -
pp.plot(x,y,linewidth=2,color='red') #Dibujamos las lineas
pp.scatter(x,y,color='blue') # Dibujamos los puntos (x,y)
pp.title("Vehiculos Atendndidos X Tiempo")
pp.grid(True) #Generamos una cuadrícula
pp.show() #Mostramos el grafico
```



Paremetrizando la simulacion

- 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 de la simulación con 3 máquinas de lavado

```

In [8]: import simpy
import random
import matplotlib.pyplot as plt

# 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 = 10
# Tiempo de simulación
TIEMPO_SIMULACION = 100

# 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)
        # Solo agregamos al diccionario al momento que termina de lavar el vehiculo.
        k = automotor[0]
        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 {0}% suciedad vehiculo => %s ' % (random.randint(30, 90)), v

    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)

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
        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(2):
        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
        rnd = random.randint(1,100)
        if rnd <= 20: # Bus 20%, Furgoneta 30% y Automovil 50%
            env.process(llegada_vehiculo(env, 'Bus-%d'%(i+1), lavanderia))
        elif rnd <= 50:
            env.process(llegada_vehiculo(env, 'Furgoneta-%d'%(i+1), lavanderia))
        else:
            env.process(llegada_vehiculo(env, 'Automovil-%d'%(i+1), lavanderia))

    print('Lavanderia UPS')

    # 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)

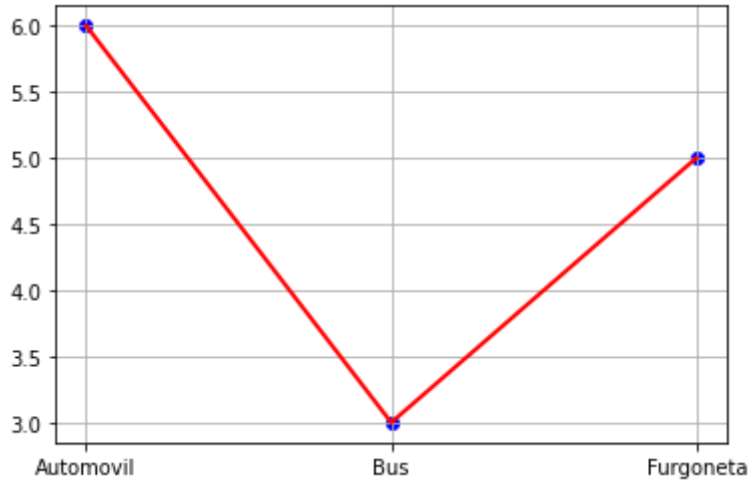
    # Generamos la grafica
    datos=sorted(tiempo.items())
    x, y =zip(*datos)
    pp.plot(x,y,linewidth=2,color='red')
    pp.scatter(x,y,color='blue')

```

```
pp.grid(True)
pp.show()
```

```
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.
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 {68%} 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-3 a la hora 11.00.
Entra vehiculo a lavarse: Bus-2 a la hora 12.00.
Removido {68%} 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: Furgoneta-4 a la hora 21.00.
Removido {63%} suciedad vehiculo => Automovil-2
Vehiculo [Automovil-2] lavado a las 22.00.
Se traslada vehiculo: Furgoneta-2 a la hora 22.00.
Removido {50%} 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.
Entra vehiculo a lavarse: Bus-3 a la hora 28.00.
Llega vehiculo: Automovil-5 a la hora 29.00.
Removido {44%} suciedad vehiculo => Bus-2
Vehiculo [Bus-2] lavado a las 32.00.
Se traslada vehiculo: Furgoneta-4 a la hora 32.00.
Entra vehiculo a lavarse: Furgoneta-4 a la hora 34.00.
Removido {67%} suciedad vehiculo => Furgoneta-2
Llega vehiculo: Furgoneta-6 a la hora 36.00.
Vehiculo [Furgoneta-2] lavado a las 36.00.
Se traslada vehiculo: Automovil-5 a la hora 36.00.
Entra vehiculo a lavarse: Automovil-5 a la hora 37.00.
Removido {76%} suciedad vehiculo => Automovil-5
Vehiculo [Automovil-5] lavado a las 44.00.
Se traslada vehiculo: Furgoneta-6 a la hora 44.00.
Llega vehiculo: Furgoneta-7 a la hora 45.00.
Removido {36%} suciedad vehiculo => Furgoneta-4
Vehiculo [Furgoneta-4] lavado a las 46.00.
Entra vehiculo a lavarse: Furgoneta-6 a la hora 46.00.
Se traslada vehiculo: Furgoneta-7 a la hora 46.00.
Removido {41%} suciedad vehiculo => Bus-3
Vehiculo [Bus-3] lavado a las 48.00.
Entra vehiculo a lavarse: Furgoneta-7 a la hora 48.00.
Llega vehiculo: Automovil-8 a la hora 57.00.
Se traslada vehiculo: Automovil-8 a la hora 57.00.
Removido {78%} suciedad vehiculo => Furgoneta-6
Vehiculo [Furgoneta-6] lavado a las 58.00.
Entra vehiculo a lavarse: Automovil-8 a la hora 58.00.
Removido {33%} suciedad vehiculo => Furgoneta-7
Vehiculo [Furgoneta-7] lavado a las 60.00.
Removido {76%} suciedad vehiculo => Automovil-8
Vehiculo [Automovil-8] lavado a las 65.00.
Llega vehiculo: Automovil-9 a la hora 68.00.
Se traslada vehiculo: Automovil-9 a la hora 68.00.
Entra vehiculo a lavarse: Automovil-9 a la hora 69.00.
Removido {87%} suciedad vehiculo => Automovil-9
Vehiculo [Automovil-9] lavado a las 76.00.
```

Llega vehiculo: Bus-10 a la hora 77.00.
 Se traslada vehiculo: Bus-10 a la hora 77.00.
 Entra vehiculo a lavarse: Bus-10 a la hora 81.00.
 Llega vehiculo: Automovil-11 a la hora 85.00.
 Se traslada vehiculo: Automovil-11 a la hora 85.00.
 Entra vehiculo a lavarse: Automovil-11 a la hora 86.00.
 Removido {68%} suciedad vehiculo => Automovil-11
 Vehiculo [Automovil-11] lavado a las 93.00.
 Llega vehiculo: Bus-12 a la hora 96.00.
 Se traslada vehiculo: Bus-12 a la hora 96.00.



Práctica SymPy 2

En base a la simulacion, generar una simulacion lo mas apegado a la realidad del Covid-19, para ello obtener informacion del numero de consultorios y camas disponibles en un hospital del Ecuador. Parametrizar la simulacion para ingresar el numero de pacientes, el numero de consultorios y el numero de camas y algun otro parametro que considere oportuno.

Finalmente generar las siguientes metricas :

- Tiempo de espera promedio para cada sala.
- Tiempo promedio total que requiere un paciente para ser atendido desde cero.
- Cuántos pacientes se atendieron.
- Cuántos pacientes se quedaron sin atender y fallecieron.

```
In [9]: import simpy
import random
import matplotlib.pyplot as pp
import numpy as np
from time import sleep
from progress.bar import Bar
%matplotlib inline
from tqdm.notebook import trange, tqdm
from time import sleep
```

```
In [13]: num_pacientes = 200
camas = 30
tiempo_atencion = 10
intervalo_llegada = 5
tiempo_simulacion = 60

tiempo = {}

class Consultorio(object):
```



```

def __init__(self, environment, camas, tiempo_atencion):
    self.env = environment
    self.camas = simpy.Resource(environment, camas)
    self.tiempo_atencion = tiempo_atencion

def atender_paciente(self, paciente):

    yield self.env.timeout(random.randint(tiempo_atencion - 8, tiempo_atencion + 8))

    print('Atendido al paciente:', paciente)

def llegada_paciente(env, nombre, Consultorio):
    print('Llega paciente: ', nombre, 'a la hora: ', env.now, 'al Hospital Bolivia por Covid-19')

    with Consultorio.camas.request() as maquina:
        yield maquina
        print('Entra paciente', nombre, 'Hora de atención: ', env.now)
        # Procesamos la operacion de atencion
        yield env.process(Consultorio.atender_paciente(nombre))
        print('paciente: ', nombre, 'atendido a las: ', env.now)

    tiempo[nombre] = env.now

def ejecutar_simulacion(env, camas, tiempo_atencion, intervalo):
    consultorio = Consultorio(env, camas, tiempo_atencion)
    for i in range(5):
        env.process(llegada_paciente(env, 'paciente-%d' % (i + 1), consultorio))

    # Ejecutamos la simulacion
    while True:
        yield env.timeout(random.randint(intervalo - 3, intervalo + 3))
        i += 1
        env.process(llegada_paciente(env, 'paciente-%d' % (i + 1), consultorio))

print('Consultorio Bolivia')

# Creamos el entorno de simulacion
env = simpy.Environment()
env.process(ejecutar_simulacion(env, camas, tiempo_atencion, intervalo_llegada))

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

print("Tiempo:")
print(tiempo)

```

Consultorio Bolivia

```

Llega paciente:  paciente-1 a la hora:  0 al Hospital Bolivia por Covid-19
Llega paciente:  paciente-2 a la hora:  0 al Hospital Bolivia por Covid-19
Llega paciente:  paciente-3 a la hora:  0 al Hospital Bolivia por Covid-19
Llega paciente:  paciente-4 a la hora:  0 al Hospital Bolivia por Covid-19
Llega paciente:  paciente-5 a la hora:  0 al Hospital Bolivia por Covid-19
Entra paciente paciente-1 Hora de atención:  0
Entra paciente paciente-2 Hora de atención:  0
Entra paciente paciente-3 Hora de atención:  0
Entra paciente paciente-4 Hora de atención:  0
Entra paciente paciente-5 Hora de atención:  0
Atendido al paciente: paciente-1
paciente:  paciente-1 atendido a las:  5
Llega paciente:  paciente-6 a la hora:  7 al Hospital Bolivia por Covid-19
Entra paciente paciente-6 Hora de atención:  7
Atendido al paciente: paciente-2

```



```

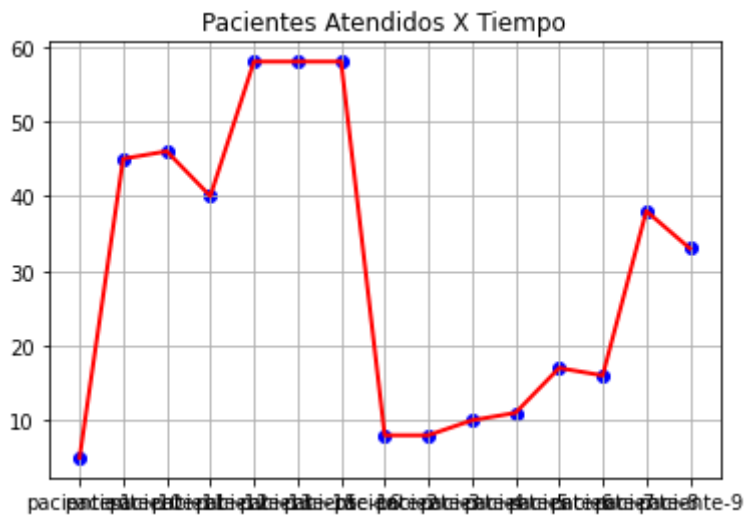
Atendido al paciente: paciente-3
paciente: paciente-2 atendido a las: 8
paciente: paciente-3 atendido a las: 8
Atendido al paciente: paciente-4
paciente: paciente-4 atendido a las: 10
Atendido al paciente: paciente-5
paciente: paciente-5 atendido a las: 11
Llega paciente: paciente-7 a la hora: 12 al Hospital Bolivia por Covid-19
Entra paciente paciente-7 Hora de atención: 12
Atendido al paciente: paciente-7
paciente: paciente-7 atendido a las: 16
Atendido al paciente: paciente-6
paciente: paciente-6 atendido a las: 17
Llega paciente: paciente-8 a la hora: 20 al Hospital Bolivia por Covid-19
Entra paciente paciente-8 Hora de atención: 20
Llega paciente: paciente-9 a la hora: 23 al Hospital Bolivia por Covid-19
Entra paciente paciente-9 Hora de atención: 23
Llega paciente: paciente-10 a la hora: 28 al Hospital Bolivia por Covid-19
Entra paciente paciente-10 Hora de atención: 28
Atendido al paciente: paciente-9
Llega paciente: paciente-11 a la hora: 33 al Hospital Bolivia por Covid-19
paciente: paciente-9 atendido a las: 33
Entra paciente paciente-11 Hora de atención: 33
Atendido al paciente: paciente-8
Llega paciente: paciente-12 a la hora: 38 al Hospital Bolivia por Covid-19
paciente: paciente-8 atendido a las: 38
Entra paciente paciente-12 Hora de atención: 38
Atendido al paciente: paciente-12
paciente: paciente-12 atendido a las: 40
Llega paciente: paciente-13 a la hora: 43 al Hospital Bolivia por Covid-19
Entra paciente paciente-13 Hora de atención: 43
Atendido al paciente: paciente-10
paciente: paciente-10 atendido a las: 45
Atendido al paciente: paciente-11
paciente: paciente-11 atendido a las: 46
Llega paciente: paciente-14 a la hora: 50 al Hospital Bolivia por Covid-19
Entra paciente paciente-14 Hora de atención: 50
Llega paciente: paciente-15 a la hora: 52 al Hospital Bolivia por Covid-19
Entra paciente paciente-15 Hora de atención: 52
Llega paciente: paciente-16 a la hora: 55 al Hospital Bolivia por Covid-19
Entra paciente paciente-16 Hora de atención: 55
Atendido al paciente: paciente-13
Atendido al paciente: paciente-15
Atendido al paciente: paciente-16
paciente: paciente-13 atendido a las: 58
paciente: paciente-15 atendido a las: 58
paciente: paciente-16 atendido a las: 58
Tiempo:
{'paciente-1': 5, 'paciente-2': 8, 'paciente-3': 8, 'paciente-4': 10, 'paciente-5':
11, 'paciente-7': 16, 'paciente-6': 17, 'paciente-9': 33, 'paciente-8': 38, 'pacient
e-12': 40, 'paciente-10': 45, 'paciente-11': 46, 'paciente-13': 58, 'paciente-15': 5
8, 'paciente-16': 58}

```

```

In [14]: # Generamos La grafica
datos=sorted(tiempo.items()) # Ordenamos Los datos
x, y =zip(*datos) # Obtener x(tiempo - clave) y el y(Numero de vehiculos atendidos -
pp.plot(x,y,linewidth=2,color='red') #Dibujamos Las lineas
pp.scatter(x,y,color='blue') # Dibujamos Los puntos (x,y)
pp.title("Pacientes Atendidos X Tiempo")
pp.grid(True) #Generamos una cuadrícula
pp.show() #Mostramos el grafico

```



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>