

simulador 3 maquinas

```
In [1]: 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 de traslado de vehiculos
TRASLADO= [4,1,2]
# Tiempo que tarda en lavarse un vehiculo (minutos)
#Bus: 20min, Automóvil: 7min, Furgoneta: 12min
TIEMPO_LAVADO =[20,7,12] #7
# Intervalo de tiempo en que llegan vehiculos (minutos)
# Bus: 4min, Automóvil: 1 min, Furgoneta: 2 min.
INTERVALO_LLEGADA = 10
# Tiempo de simulación
TIEMPO_SIMULACION = 100

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(TRASLADO[0])
        if(automotor[0] == 'Automovil'):
            yield self.env.timeout(TRASLADO[1])
        if(automotor[0] == 'Furgoneta'):
            yield self.env.timeout(TRASLADO[2])
        #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 {%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[0])
        if(automotor[0] == 'Automovil'):
            yield self.env.timeout(TRASLADO[1])
        if(automotor[0] == 'Furgoneta'):
            yield self.env.timeout(TRASLADO[2])
```

```

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(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')

# Inicializamos la semilla aleatoria
random.seed(77)

# Creamos el entorno de simulacion
env=simpy.Environment()
env.process(ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO[0], TIEMPO_LAVADO[1], \
                                TIEMPO_LAVADO[2] \
                                , TRASLADO[0], TRASLADO[1], TRASLADO[2] \
                                , INTERVALO_LLEGADA))

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

```

```

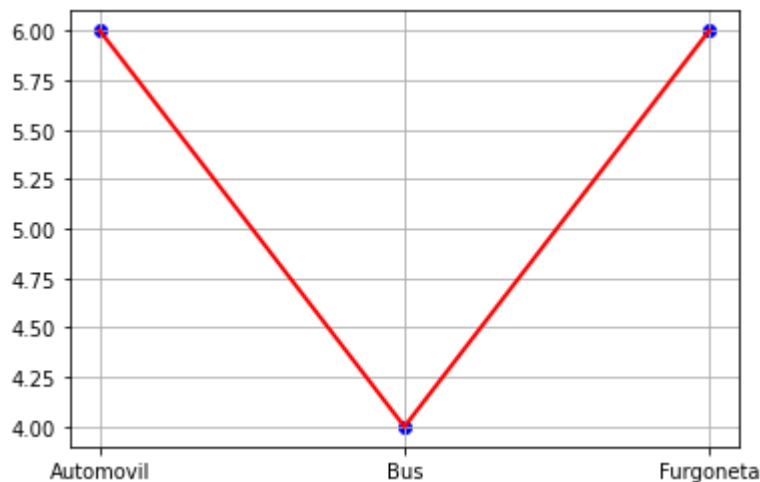
Lavanderia UPS
Llega vehiculo: Bus-1 a la hora 0.00.
Llega vehiculo: Automovil-1 a la hora 0.00.
Llega vehiculo: Furgoneta-1 a la hora 0.00.
Llega vehiculo: Bus-2 a la hora 0.00.
Llega vehiculo: Automovil-2 a la hora 0.00.
Llega vehiculo: Furgoneta-2 a la hora 0.00.
Se traslada vehiculo: Bus-1 a la hora 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.
Removido {46%} suciedad vehiculo => Automovil-1
Entra vehiculo a lavarse: Furgoneta-1 a la hora 2.00.
Vehiculo [Automovil-1] lavado a las 2.00.
Se traslada vehiculo: Bus-2 a la hora 2.00.
Removido {50%} suciedad vehiculo => Furgoneta-1
Entra vehiculo a lavarse: Bus-1 a la hora 4.00.
Vehiculo [Furgoneta-1] lavado a las 4.00.
Se traslada vehiculo: Automovil-2 a la hora 4.00.
Entra vehiculo a lavarse: Automovil-2 a la hora 5.00.
Removido {42%} suciedad vehiculo => Automovil-2
Entra vehiculo a lavarse: Bus-2 a la hora 6.00.
Vehiculo [Automovil-2] lavado a las 6.00.
Se traslada vehiculo: Furgoneta-2 a la hora 6.00.
Removido {45%} suciedad vehiculo => Bus-1
Vehiculo [Bus-1] lavado a las 8.00.
Entra vehiculo a lavarse: Furgoneta-2 a la hora 8.00.
Removido {42%} suciedad vehiculo => Bus-2
Removido {82%} suciedad vehiculo => Furgoneta-2
Vehiculo [Bus-2] lavado a las 10.00.
Vehiculo [Furgoneta-2] lavado a las 10.00.
Llega vehiculo: Bus-3 a la hora 13.00.
Se traslada vehiculo: Bus-3 a la hora 13.00.
Entra vehiculo a lavarse: Bus-3 a la hora 17.00.
Removido {60%} suciedad vehiculo => Bus-3
Vehiculo [Bus-3] lavado a las 21.00.
Llega vehiculo: Automovil-4 a la hora 22.00.
Se traslada vehiculo: Automovil-4 a la hora 22.00.
Entra vehiculo a lavarse: Automovil-4 a la hora 23.00.
Removido {45%} suciedad vehiculo => Automovil-4
Vehiculo [Automovil-4] lavado a las 24.00.
Llega vehiculo: Bus-5 a la hora 33.00.
Se traslada vehiculo: Bus-5 a la hora 33.00.
Entra vehiculo a lavarse: Bus-5 a la hora 37.00.
Removido {70%} suciedad vehiculo => Bus-5
Vehiculo [Bus-5] lavado a las 41.00.
Llega vehiculo: Automovil-6 a la hora 44.00.
Se traslada vehiculo: Automovil-6 a la hora 44.00.
Entra vehiculo a lavarse: Automovil-6 a la hora 45.00.
Removido {30%} suciedad vehiculo => Automovil-6
Vehiculo [Automovil-6] lavado a las 46.00.
Llega vehiculo: Furgoneta-7 a la hora 55.00.
Se traslada vehiculo: Furgoneta-7 a la hora 55.00.
Entra vehiculo a lavarse: Furgoneta-7 a la hora 57.00.
Removido {83%} suciedad vehiculo => Furgoneta-7
Vehiculo [Furgoneta-7] lavado a las 59.00.
Llega vehiculo: Automovil-8 a la hora 62.00.
Se traslada vehiculo: Automovil-8 a la hora 62.00.
Entra vehiculo a lavarse: Automovil-8 a la hora 63.00.
Removido {80%} suciedad vehiculo => Automovil-8
Vehiculo [Automovil-8] lavado a las 64.00.
Llega vehiculo: Furgoneta-9 a la hora 70.00.
Se traslada vehiculo: Furgoneta-9 a la hora 70.00.
Entra vehiculo a lavarse: Furgoneta-9 a la hora 72.00.
Removido {83%} suciedad vehiculo => Furgoneta-9
Vehiculo [Furgoneta-9] lavado a las 74.00.
Llega vehiculo: Furgoneta-10 a la hora 81.00.
Se traslada vehiculo: Furgoneta-10 a la hora 81.00.
Entra vehiculo a lavarse: Furgoneta-10 a la hora 83.00.
Removido {51%} suciedad vehiculo => Furgoneta-10
Vehiculo [Furgoneta-10] lavado a las 85.00.
Llega vehiculo: Furgoneta-11 a la hora 89.00.
Se traslada vehiculo: Furgoneta-11 a la hora 89.00.
Entra vehiculo a lavarse: Furgoneta-11 a la hora 91.00.
Removido {35%} suciedad vehiculo => Furgoneta-11
Vehiculo [Furgoneta-11] lavado a las 93.00.
Llega vehiculo: Automovil-12 a la hora 96.00.
Se traslada vehiculo: Automovil-12 a la hora 96.00.
Entra vehiculo a lavarse: Automovil-12 a la hora 97.00.
Removido {46%} suciedad vehiculo => Automovil-12
Vehiculo [Automovil-12] lavado a las 98.00.

grafica

```
In [2]: datos = sorted(tiempo.items())
x, y = zip(*datos)
plt.plot(x,y,linewidth=2, color="red")
plt.scatter(x,y,color='blue')
plt.grid(True)
plt.show()
```



4 maquinas

librerias

```
In [3]: import simpy
import random
import matplotlib.pyplot as plt
```

variables

```
In [4]: # 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 de traslado de vehiculos
TRASLADO= [4,1,2]
# Tiempo que tarda en lavarse un vehiculo (minutos)
#Bus: 20min, Automóvil: 7min, Furgoneta: 12min
TIEMPO_LAVADO = [20,7,12] #7
# Intervalo de tiempo en que llegan vehiculos (minutos)
# Bus: 4min, Automóvil: 1 min, Furgoneta: 2 min.
INTERVALO_LLEGADA = 10
# Tiempo de simulación
TIEMPO_SIMULACION = 100

tiempo= {}
```

clases

```
In [5]: 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(TRASLADO[0])
    if(automotor[0] == 'Automovil'):
        yield self.env.timeout(TRASLADO[1])
    if(automotor[0] == 'Furgoneta'):
        yield self.env.timeout(TRASLADO[2])
    #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 {0d}% 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[0])
    if(automotor[0] == 'Automovil'):
        yield self.env.timeout(TRASLADO[1])
    if(automotor[0] == 'Furgoneta'):
        yield self.env.timeout(TRASLADO[2])

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 {0.2f}.' % (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 {0.2f}.' % (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 {0.2f}.' % (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 {0.2f}.' % (nombre, env.now))

```

ejecutar

```

In [6]: 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-{0d}'%(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')
```

Lavanderia UPS

In [7]:

```
# Inicializamos la semilla aleatoria
random.seed(80)

# Creamos el entorno de simulacion
env=simpy.Environment()
env.process(ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO[0],TIEMPO_LAVADO[1],\
                                TIEMPO_LAVADO[2]\
                                ,TRASLADO[0],TRASLADO[1],TRASLADO[2]
                                , INTERVALO_LLEGADA))

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

```
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.
Se traslada vehiculo: Bus-2 a la hora 0.00.
Entra vehiculo a lavarse: Automovil-1 a la hora 1.00.
Removido {55%} suciedad vehiculo => Automovil-1
Entra vehiculo a lavarse: Furgoneta-1 a la hora 2.00.
Vehiculo [Automovil-1] lavado a las 2.00.
Se traslada vehiculo: Automovil-2 a la hora 2.00.
Entra vehiculo a lavarse: Automovil-2 a la hora 3.00.
Removido {64%} suciedad vehiculo => Furgoneta-1
Removido {87%} suciedad vehiculo => Automovil-2
Entra vehiculo a lavarse: Bus-1 a la hora 4.00.
Entra vehiculo a lavarse: Bus-2 a la hora 4.00.
Vehiculo [Furgoneta-1] lavado a las 4.00.
Vehiculo [Automovil-2] lavado a las 4.00.
Se traslada vehiculo: Furgoneta-2 a la hora 4.00.
Entra vehiculo a lavarse: Furgoneta-2 a la hora 6.00.
Removido {75%} suciedad vehiculo => Bus-1
Removido {85%} suciedad vehiculo => Bus-2
Removido {56%} suciedad vehiculo => Furgoneta-2
Vehiculo [Bus-1] lavado a las 8.00.
Vehiculo [Bus-2] lavado a las 8.00.
Vehiculo [Furgoneta-2] lavado a las 8.00.
Llega vehiculo: Furgoneta-3 a la hora 9.00.
Se traslada vehiculo: Furgoneta-3 a la hora 9.00.
Entra vehiculo a lavarse: Furgoneta-3 a la hora 11.00.
Removido {64%} suciedad vehiculo => Furgoneta-3
Vehiculo [Furgoneta-3] lavado a las 13.00.
Llega vehiculo: Automovil-4 a la hora 18.00.
Se traslada vehiculo: Automovil-4 a la hora 18.00.
Entra vehiculo a lavarse: Automovil-4 a la hora 19.00.
Removido {31%} suciedad vehiculo => Automovil-4
Vehiculo [Automovil-4] lavado a las 20.00.
```

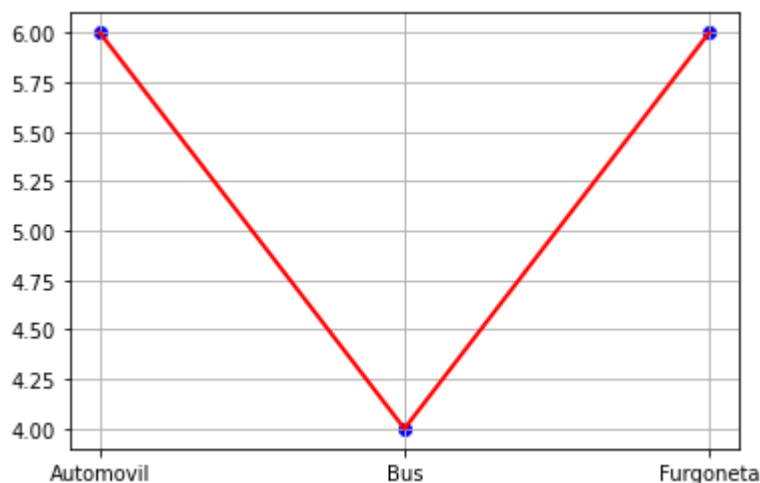
Llega vehiculo: Automovil-5 a la hora 27.00.
 Se traslada vehiculo: Automovil-5 a la hora 27.00.
 Entra vehiculo a lavarse: Automovil-5 a la hora 28.00.
 Removido {83%} suciedad vehiculo => Automovil-5
 Vehiculo [Automovil-5] lavado a las 29.00.
 Llega vehiculo: Furgoneta-6 a la hora 40.00.
 Se traslada vehiculo: Furgoneta-6 a la hora 40.00.
 Entra vehiculo a lavarse: Furgoneta-6 a la hora 42.00.
 Removido {90%} suciedad vehiculo => Furgoneta-6
 Vehiculo [Furgoneta-6] lavado a las 44.00.
 Llega vehiculo: Automovil-7 a la hora 48.00.
 Se traslada vehiculo: Automovil-7 a la hora 48.00.
 Entra vehiculo a lavarse: Automovil-7 a la hora 49.00.
 Removido {54%} suciedad vehiculo => Automovil-7
 Vehiculo [Automovil-7] lavado a las 50.00.
 Llega vehiculo: Furgoneta-8 a la hora 55.00.
 Se traslada vehiculo: Furgoneta-8 a la hora 55.00.
 Entra vehiculo a lavarse: Furgoneta-8 a la hora 57.00.
 Removido {90%} suciedad vehiculo => Furgoneta-8
 Vehiculo [Furgoneta-8] lavado a las 59.00.
 Llega vehiculo: Automovil-9 a la hora 63.00.
 Se traslada vehiculo: Automovil-9 a la hora 63.00.
 Entra vehiculo a lavarse: Automovil-9 a la hora 64.00.
 Removido {30%} suciedad vehiculo => Automovil-9
 Vehiculo [Automovil-9] lavado a las 65.00.
 Llega vehiculo: Bus-10 a la hora 74.00.
 Se traslada vehiculo: Bus-10 a la hora 74.00.
 Entra vehiculo a lavarse: Bus-10 a la hora 78.00.
 Removido {69%} suciedad vehiculo => Bus-10
 Vehiculo [Bus-10] lavado a las 82.00.
 Llega vehiculo: Bus-11 a la hora 84.00.
 Se traslada vehiculo: Bus-11 a la hora 84.00.
 Entra vehiculo a lavarse: Bus-11 a la hora 88.00.
 Llega vehiculo: Furgoneta-12 a la hora 92.00.
 Removido {52%} suciedad vehiculo => Bus-11
 Se traslada vehiculo: Furgoneta-12 a la hora 92.00.
 Vehiculo [Bus-11] lavado a las 92.00.
 Entra vehiculo a lavarse: Furgoneta-12 a la hora 94.00.
 Removido {67%} suciedad vehiculo => Furgoneta-12
 Vehiculo [Furgoneta-12] lavado a las 96.00.

grafica

```

In [8]: datos2 = sorted(tiempo.items())
x, y = zip(*datos2)
plt.plot(x,y,linewidth=2, color="red")
plt.scatter(x,y,color='blue')
plt.grid(True)
plt.show()

```



In []:

In []:

