

Simulación de eventos discretos - Car wash

Maestro: Diego Quisi

Autor: Bryam David Vega Moreno

Materia: Simulación Universidad: Universidad Politécnica Salesiana Carrera: Ciencias de la computación

Introducción

En el presente trabajo se procedera a realizar la simulación de un evento a traves de una ejemplificación de lavado de carros mediante máquinas con la finalidad de entender como funciona la simulación con simpy y sus procesos. **Enunciado** Este ejemplo permite simular un negocio de car wash. De igual forma, es importante destacar que este ejemplo aborda los siguientes puntos: Estados de espera por otros procesos

• Recursos: clase Resource Características del sistema real a simular Es importante observar que el negocio de lavado de autos tiene características propias a su naturaleza. A continuación establecemos dichas peculiaridades a tener en mente en el momento de llevar a cabo la simulación:

• El negocio tiene un número limitado de máquinas de lavado. • Se puede recibir un cierto número de vehículos para ser lavados, de los cuáles los que no estén siendo procesados tendrán que esperar. vehículo.

Desarrollo de la simulación En este desarrollo tomaremo en cuenta las variables de interes que necesita nuestro sistema de simulación para que trabaje correctamente, de la misma

manera realizaremos la programación explicando cada uno de los procesos.

Importar librerías Importamos las librerías necesarias para desarrollar el proyecto de simulación import random import simpy

In [1]: import numpy Variables de interes En este caso hemos decidido utilizar las siguientes variables de interes

• máximo de vehiculos: Numero máximo de vehiculos que puede recibir el local • tiempo de simulación: Tiempo en que se tardara la simulación • tiempo de lavado: Tiempo de lavado del vehículo

• intervalo de llegada: Intervalo del tiempo en el que llegan vehiculos • número de máquinas: Máquinas que proceden a realizar el lavado al vehículo Con estas variables de interes procedemos a realizar el proceso de inicializarlas con valores:

In [2]: MAX_VEHICULOS = 57 NUM_MAQUINAS TIEMPO_LAVADO = 7 INTERVALO_LLEGADA = 9 $TIEMPO_SIMULACION = 23$

Desarrollar las clases para la simulación Para esta práctica vamos a contrar con las siguientes clases:

• Lavanderia : Ejecutra los procesos nomrales de la lavandería • Simulacion : Ejecuta el proceso de simulación

In [3]: class Lavanderia(): def __init__ (self,enviroment,num_maquinas,tiempo_lavado): self.env = enviroment

self.maquinas = simpy.Resource(environment, num_maquinas) self.tiempo_lavado = tiempo_lavado def lavar_vehiculo(self, vehiculo): yield self.env.timeout(TIEMPO_LAVADO) print("Lavando el vehículo {}".format(vehículo)) def llegada_vehiculo(self, vehiculo): print('Llega vehiculo: %s a la hora %.2f.' % (vehiculo, self.env.now)) with self.maguinas.request() as maguina: print('Entra vehiculo a lavarse: %s a la hora %.2f.' % (vehiculo, env.now))

yield env.process(self.lavar_vehiculo(vehiculo)) print('Vehiculo [%s] lavado a las %.2f.' % (vehiculo, env.now)) class Simulacion(): def __init__(self,inicial): self.inicial = inicial def ejecutar_simulacion(self,env, num_maquinas, tiempo_lavado, intervalo): lavanderia=Lavanderia(env, num_maguinas, tiempo_lavado) self.inicializar_vehiculos(env,lavanderia) while True: vield env.timeout(random.randint(intervalo-2, intervalo+2)) self.inicial+=1 env.process(lavanderia.llegada_vehiculo('Vehiculo-%d'%(self.inicial))) def inicializar_vehiculos(self, env, lavanderia): for i in range(self.inicial): env.process(lavanderia.llegada_vehiculo('Vehiculo-%d'%(i+1))) In [4]: print('*'*10,'Lavanderia Vega Simulación','*'*10) random.seed(77)

env.process(simulacion.ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO, INTERVALO_LLEGADA))

env=simpy.Environment() simulacion = Simulacion(3)

In [5]: class Lavanderia():

self.env = enviroment

def lavar_vehiculo(self, vehiculo):

self.tiempo_lavado = tiempo_lavado

yield self.env.timeout(TIEMPO_LAVADO)

def llegada_al_lavado(self, vehiculo, espera):

yield self.env.timeout(espera)

def llegada_vehiculo(self, vehiculo):

Lavando el vehículo Vehiculo-1

Vehiculo [Vehiculo-1] lavado a las 10.00.

Vehiculo empieza transporte: Vehiculo-4 a la hora 9.00. Entra vehiculo a lavarse: Vehiculo-4 a la hora 10.00. Vehiculo empieza transporte: Vehiculo-5 a la hora 9.00. Entra vehiculo a lavarse: Vehiculo-5 a la hora 12.00.

def llegada_al_lavado(self, vehiculo, espera):

for maquina in self.maquinas_uso.items():

Vehiculo [Vehiculo-2] lavado en la machine_2 a las 9.00. Vehiculo [Vehiculo-3] lavado en la machine_3 a las 9.00.

Vehiculo [Vehiculo-1] lavado en la machine_1 a las 10.00.

Vehiculo [Vehiculo-4] lavado en la machine_2 a las 17.00.

def __init__ (self,enviroment,num_maquinas,tiempo_lavado):

print("Lavando el vehículo {}".format(vehículo))

self.maquinas = simpy.Resource(environment, num_maquinas)

Entra vehiculo a lavarse: Vehiculo-4 a la hora 10.00.

Entra vehiculo a lavarse: Vehiculo-5 a la hora 21.00.

Vehiculo empieza transporte en la maquina machine_2: Vehiculo-4 a la hora 9.00.

Vehiculo empieza transporte en la maquina machine_1: Vehiculo-5 a la hora 17.00.

Lavando el vehículo Vehiculo-1

Lavando el vehículo Vehiculo-4

self.env = enviroment

def lavar_vehiculo(self, vehiculo):

self.tiempo_lavado = tiempo_lavado

yield self.env.timeout(TIEMPO_LAVADO)

def llegada_al_lavado(self, vehiculo, espera):

yield self.env.timeout(espera)

******* Lavanderia Vega Simulación *******

Entra vehiculo a lavarse: Vehiculo-2 a la hora 2.00.

Entra vehiculo a lavarse: Vehiculo-3 a la hora 2.00.

Entra vehiculo a lavarse: Vehiculo-1 a la hora 3.00.

Vehiculo sale : Vehiculo-2 a la hora 11.00.

Vehiculo sale : Vehiculo-3 a la hora 13.00.

Vehiculo sale : Vehiculo-1 a la hora 15.00.

Modificar el tiempo de lavado entre 5 y 11

self.env = enviroment

def lavar_vehiculo(self, vehiculo):

Vehiculo sale : Vehiculo-2 a la hora 12.00.

Vehiculo sale : Vehiculo-1 a la hora 13.00.

Vehiculo sale : Vehiculo-3 a la hora 16.00.

refactorizados y entendibles para cualquier desarrollador.

Proceso de lavado del vehículo

Proceso de llegada a la máquina

Proceso de salida del carwash

def vacate_machine(self, key):

def arrived_car(self,car):

Proceso del arrivo del vehículo

def arrived_machine(self):

def go_out_carwash(self):

def __init__(self, environment, machines):

Vehiculo [Vehiculo-3] lavado en la machine_3 a las 13.00.

Entra vehiculo a lavarse: Vehiculo-4 a la hora 17.00.

a realizar una refactorización a fin de dejar listo el código y sea facil de entender.

Vehiculo empieza transporte a la salida : Vehiculo-1 a la hora 8.00.

Vehiculo empieza transporte a la salida : Vehiculo-3 a la hora 13.00.

Vehiculo empieza transporte en la maquina machine_1: Vehiculo-4 a la hora 12.00.

Lavando el vehículo Vehiculo-3

Refactorización

self.env self.machine

os de las máquinas usadas

def wash car(self):

In [22]: class CarWash():

self.tiempo_lavado = tiempo_lavado

clase lavandería.

In [11]: class Lavanderia():

Vehiculo [Vehiculo-2] lavado en la machine_2 a las 9.00. Vehiculo [Vehiculo-3] lavado en la machine_3 a las 9.00.

Vehiculo [Vehiculo-1] lavado en la machine_1 a las 10.00.

Entra vehiculo a lavarse: Vehiculo-4 a la hora 16.00.

Entra vehiculo a lavarse: Vehiculo-5 a la hora 19.00.

del vehículo, por lo que es necesario poner un randomico en el tiempo de lavado.

def __init__ (self,enviroment,num_maquinas,tiempo_lavado):

yield self.env.timeout(random.randint(5,11))

self.maquinas = simpy.Resource(environment, num_maquinas)

Vehiculo empieza transporte a la salida : Vehiculo-2 a la hora 9.00.

Vehiculo empieza transporte a la salida : Vehiculo-3 a la hora 9.00.

Vehiculo empieza transporte a la salida : Vehiculo-1 a la hora 10.00.

Lavando el vehículo Vehiculo-2 Lavando el vehículo Vehiculo-3

Lavando el vehículo Vehiculo-1

Vehiculo empieza transporte en la maquina machine_2: Vehiculo-2 a la hora 0.00.

Vehiculo empieza transporte en la maquina machine_3: Vehiculo-3 a la hora 0.00.

Vehiculo empieza transporte en la maquina machine_1: Vehiculo-1 a la hora 0.00.

Vehiculo empieza transporte en la maquina machine_1: Vehiculo-4 a la hora 11.00.

Vehiculo empieza transporte en la maquina machine_2: Vehiculo-5 a la hora 17.00.

Como podemos darnos cuenta, ahora se tiene simulado la salida del local del los vehículos, con esto hemos detallado mucho más nuestra simulación, ahora nos faltaría realizar una simulación del tiempo aleatorio del lavado ya que estamos poniendo tiempos iguales, sin embargo, esto podría varias dependiendo

En esta ocasión vamos a realizar un ultimo ajuste a nuestra simulación realizando un cambio en el tiempo de lavado. Para ello realizamos el cambio en la

self.maquinas_uso = {'machine_{{}}'.format(i+1):False for i in range(num_maquinas)}

def salida_local(self, espera):

In [9]: class Lavanderia():

yield self.env.timeout(espera)

if(maquina[1]==False):

def usar_maquina(self): maquina_nombre=''

env.run(until = TIEMPO_SIMULACION)

******* Lavanderia Vega Simulación *******

def __init__ (self, environment, num_maquinas, tiempo_lavado):

print("Lavando el vehículo {}".format(vehículo))

self.maquinas = simpy.Resource(environment, num_maquinas)

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. 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. Lavando el vehículo Vehiculo-1 Lavando el vehículo Vehiculo-2 Lavando el vehículo 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. Llega vehiculo: Vehiculo-4 a la hora 9.00. Entra vehiculo a lavarse: Vehiculo-4 a la hora 9.00. Lavando el vehículo Vehiculo-4 Vehiculo [Vehiculo-4] lavado a las 16.00. Llega vehiculo: Vehiculo-5 a la hora 18.00. Entra vehiculo a lavarse: Vehiculo-5 a la hora 18.00. Como podemos apreciar, esta la simulación básica de un lavado de carro, ahora, vamos a realizar otros procesos, con el fin de ir detallando de mejor manera la simulación y que sea un poco más real Agregar un tiempo de espera entre la llegada del vehiculo y la llegada a la maquina de lavado con un intervalo (1,5)Procedemos a crear un método que nos permita saber el tiempo que se demorará el vehículo en llegar a la máquina de lavado a partir de su llegada inicial. Por tanto debemos modificar la clase Lavandería indicando un número aleatorio de 1 a 5 con la finalidad de tener una mayor realidad en la simulación ya que los vehiculos pueden demorarse en llegar a la maquina en diferentes tiempos.

with self.maquinas.request() as maquina: yield maquina espera = random.randint(1,5)yield env.process(self.llegada_al_lavado(vehiculo, espera)) print('Vehiculo empieza transporte: %s a la hora %.2f.'% (vehiculo, env.now-(espera))) print('Entra vehiculo a lavarse: %s a la hora %.2f.' % (vehiculo, env.now)) yield env.process(self.lavar_vehiculo(vehiculo)) print('Vehiculo [%s] lavado a las %.2f.' % (vehiculo, env.now)) In [6]: print('*'*10,'Lavanderia Vega Simulación','*'*10) random.seed(77)env=simpy.Environment() simulacion = Simulacion(5) env.process(simulacion.ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO, INTERVALO_LLEGADA)) env.run(until = TIEMPO_SIMULACION) ******* Lavanderia Vega Simulación ******* Vehiculo empieza transporte: Vehiculo-2 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-2 a la hora 2.00. Vehiculo empieza transporte: Vehiculo-3 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-3 a la hora 2.00. Vehiculo empieza transporte: Vehiculo-1 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-1 a la hora 3.00. Lavando el vehículo Vehiculo-2 Lavando el vehículo Vehiculo-3 Vehiculo [Vehiculo-2] lavado a las 9.00. Vehiculo [Vehiculo-3] lavado a las 9.00.

Vehiculo empieza transporte: Vehiculo-6 a la hora 10.00. Entra vehiculo a lavarse: Vehiculo-6 a la hora 14.00. Lavando el vehículo Vehiculo-4 Vehiculo [Vehiculo-4] lavado a las 17.00. Lavando el vehículo Vehiculo-5 Vehiculo [Vehiculo-5] lavado a las 19.00. Lavando el vehículo Vehiculo-6 Vehiculo [Vehiculo-6] lavado a las 21.00. Vehiculo empieza transporte: Vehiculo-7 a la hora 17.00. Entra vehiculo a lavarse: Vehiculo-7 a la hora 22.00. Como podemos notar, ahora que tenemos el tiempo en el que el vehículo tarda en llegar a la máquina, podemos notar claramente como ahora los vehículos no llegan a la vez ya que su tiempo es aleatorio. Sin embargo, sería bueno identificar la máquina que cada vehīculo esta ocupando con la finalidad de tener más detalle de la simulación. Identificar el número de máquina dentro de cada proceso Es importante saber en que máquina se encuentra nuestro vehículo con la finalidad de poder saber a que máquina entra un nuevo vehículo y cual se desocupa. Para ello igualmente modificaremos la clase de Lavandería In [7]: class Lavanderia(): def __init__ (self,enviroment,num_maquinas,tiempo_lavado): self.env = enviroment self.maquinas = simpy.Resource(enviroment, num_maquinas) self.tiempo_lavado = tiempo_lavado self.maquinas_uso = {'machine_{{}}'.format(i+1):False for i in range(num_maquinas)} def lavar_vehiculo(self, vehiculo): yield self.env.timeout(TIEMPO_LAVADO) print("Lavando el vehículo {}".format(vehículo))

self.maquinas_uso[maquina[0]]=True maquina_nombre=maquina[0] break return maquina_nombre def desocupar_maquina(self, key): self.maquinas_uso[key]=False def llegada_vehiculo(self, vehiculo): with self.maquinas.request() as maquina: yield maquina maquina_nombre = self.usar_maquina() espera = random.randint(1,5) yield env.process(self.llegada_al_lavado(vehiculo, espera)) print('Vehiculo empieza transporte en la maquina %s: %s a la hora %.2f.'% (maquina_nombre, vehiculo, env. now-(espera))) print('Entra vehiculo a lavarse: %s a la hora %.2f.' % (vehiculo, env.now)) yield env.process(self.lavar_vehiculo(vehiculo)) self.desocupar_maquina(maquina_nombre) print('Vehiculo [%s] lavado en la %s a las %.2f.' % (vehiculo, maquina_nombre, env.now)) In [8]: print('*'*10,'Lavanderia Vega Simulación','*'*10) random.seed(77)env=simpy.Environment() simulacion = Simulacion(3) env.process(simulacion.ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO, INTERVALO_LLEGADA)) env.run(until = TIEMPO_SIMULACION) ****** Lavanderia Vega Simulación ******* Vehiculo empieza transporte en la maquina machine_2: Vehiculo-2 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-2 a la hora 2.00. Vehiculo empieza transporte en la maquina machine_3: Vehiculo-3 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-3 a la hora 2.00. Vehiculo empieza transporte en la maquina machine_1: Vehiculo-1 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-1 a la hora 3.00. Lavando el vehículo Vehiculo-2 Lavando el vehículo Vehiculo-3

Como nos podemos dar cuenta, ahora se puede identificar a que máquina entra cada vehículo con la finalidad de ver a más detalle lo que sucede. Sin

Agregar y modificar el tiempo de salir de la máquina a la puerta principal del negocio con un intervalo (2,5)

de espera entre la máquina y la salida, por esa razón eso se debe simular. Para ello modificamos la clase Lavanderia nuevamente

self.maquinas_uso = {'machine_{{}}'.format(i+1):False for i in range(num_maquinas)}

embargo, podemos detallar a un más, indicando el tiempo en el que el vehículo se demora en salir del local una vez que acabo de ser lavado en la máquina.

Como explicamos antes, tenemos que tomar en cuenta que una vez que el auto se acabo de lavar, tiene que salir del local y por lo tanto habrá cierto tiempo

yield self.env.timeout(espera) def usar_maquina(self): maquina_nombre='' for maquina in self.maquinas_uso.items(): if(maquina[1]==False): self.maquinas_uso[maquina[0]]=True maquina_nombre=maquina[0] break return maquina_nombre def desocupar_maquina(self, key): self.maquinas_uso[key]=False def llegada_vehiculo(self, vehiculo): with self.maquinas.request() as maquina: yield maquina maquina_nombre = self.usar_maquina() espera = random.randint(1,5) yield env.process(self.llegada_al_lavado(vehiculo, espera)) print('Vehiculo empieza transporte en la maquina %s: %s a la hora %.2f.'% (maquina_nombre, vehiculo, env. now-(espera))) print('Entra vehiculo a lavarse: %s a la hora %.2f.' % (vehiculo, env.now)) yield env.process(self.lavar_vehiculo(vehiculo)) self.desocupar_maquina(maquina_nombre) print('Vehiculo [%s] lavado en la %s a las %.2f.' % (vehiculo, maquina_nombre, env.now)) salida = random.randint(2,5) yield env.process(self.salida_local(salida)) print('Vehiculo empieza transporte a la salida : %s a la hora %.2f.'% (vehiculo, env.now-(salida))) print('Vehiculo sale : %s a la hora %.2f.'% (vehiculo, env.now)) In [10]: print('*'*10,'Lavanderia Vega Simulación','*'*10) random.seed(77)env=simpy.Environment() simulacion = Simulacion(3) env.process(simulacion.ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO, INTERVALO_LLEGADA)) env.run(until = TIEMPO_SIMULACION)

print("Lavando el vehículo {}".format(vehículo)) def llegada_al_lavado(self, vehiculo, espera): yield self.env.timeout(espera) def salida_local(self, espera): yield self.env.timeout(espera) def usar_maquina(self): maquina_nombre='' for maquina in self.maquinas_uso.items(): if(maquina[1]==False): self.maquinas_uso[maquina[0]]=True maquina_nombre=maquina[0] break return maquina_nombre def desocupar_maquina(self, key): self.maquinas_uso[key]=False def llegada_vehiculo(self, vehiculo): with self.maquinas.request() as maquina: yield maquina maquina_nombre = self.usar_maquina() espera = random.randint(1,5) yield env.process(self.llegada_al_lavado(vehiculo, espera)) print('Vehiculo empieza transporte en la maquina %s: %s a la hora %.2f.'% (maquina_nombre, vehiculo, env. now-(espera))) print('Entra vehiculo a lavarse: %s a la hora %.2f.' % (vehiculo, env.now)) yield env.process(self.lavar_vehiculo(vehiculo)) self.desocupar_maquina(maquina_nombre) print('Vehiculo [%s] lavado en la %s a las %.2f.' % (vehiculo, maquina_nombre, env.now)) salida = random.randint(2,5) vield env.process(self.salida local(salida)) print('Vehiculo empieza transporte a la salida : %s a la hora %.2f.'% (vehiculo, env.now-(salida))) print('Vehiculo sale : %s a la hora %.2f.'% (vehiculo, env.now)) In [12]: print('*'*10,'Lavanderia Vega Simulación','*'*10) random.seed(77)env=simpy.Environment() simulacion = Simulacion(3) env.process(simulacion.ejecutar_simulacion(env, NUM_MAQUINAS, TIEMPO_LAVADO, INTERVALO_LLEGADA)) env.run(until = TIEMPO_SIMULACION) ******* Lavanderia Vega Simulación ******* Vehiculo empieza transporte en la maquina machine_2: Vehiculo-2 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-2 a la hora 2.00. Vehiculo empieza transporte en la maquina machine_3: Vehiculo-3 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-3 a la hora 2.00. Vehiculo empieza transporte en la maquina machine_1: Vehiculo-1 a la hora 0.00. Entra vehiculo a lavarse: Vehiculo-1 a la hora 3.00. Lavando el vehículo Vehiculo-2 Lavando el vehículo Vehiculo-1 Vehiculo [Vehiculo-2] lavado en la machine_2 a las 8.00. Vehiculo [Vehiculo-1] lavado en la machine_1 a las 8.00. Vehiculo empieza transporte a la salida : Vehiculo-2 a la hora 8.00.

Como podemos notar ahora ya existen diferentes tiempos de lavado de autos, haciendo más real la simulación. Por ultimo para dejar todo listo, procedemos

Esto nos va a permitir entender de una manera más sencilla todo el código que hemos ido realizando a lo largo de estos ejemplos y unificarlos en uno solo.

= simpy.Resource(environment, machines) # inicializamos los recursos de nuestras máquinas

self.use_machine = {'maquina_{}'.format(i+1):False for i in range(machines)}# dicionario para guardar estad

Para esto vamos a crear las clases CarWash y Simulation que tienen los métodos realizados en los puntos anteriores, con el objetivo de que estan

= environment # inicializamos el entorno de simulación

self.out_time = 0 # timpo de salida de la máquina hacia la puerta de salida del local

yield self.env.timeout(self.wash_time)# Realizamos la simulación del lavado del vehículo

yield self.env.timeout(self.wait_time) # Realizamos la simulación de llegada a la máquina

yield self.env.timeout(self.out_time) # Realizamos la simulación de salida del carwash

self.wait_time = 0 # tiempo de espera para llegar a la máquina de lavado

self.wash_time = random.randint(5,11) # Tiempo de lavado del vehículo

self.wash_time = 0 # tiempo de lavado del vehículo

Proceso de asignar máquina a vehículo def busy_machine(self): name_machine = '' # nombre de la máquina a retornar for machine in self.use_machine.items(): # recorremos el diccionario de máquinas if(machine[1]==False): # Si la máquina tiene estado False, significa que no esta usada self.use_machine[machine[0]]=**True** # Se cambia el estado de la máquina a True, para indicar que se es ta ocupando name_machine = machine[0] # Asigno el nombre de la maquina a ocupar break return name_machine # retorno el nombre de la máquina # Proceso de desocupar máquina

self.use_machine[key] = **False** # Asignamos un valor de False para indicar que la máquina esta desalojada

print('[{}] llega a la hora {:,.2f}'.format(car, self.env.now)) with self.machine.request() as machine: # Utilizamos los recursos de máquinas de lavado yield machine # Ocupo el recurso de la máquina name_machine = self.busy_machine() # Obtengo el nombre de la máquina a ocupar # Proceso de llegada a la máquina self.wait_time = random.randint(1,5) # Tiempo de llegada hacia la máquina print('[{}] empieza a dirigirse a la [{}] a la hora {:,.2f}'.format(car,name_machine,(self.env.now))) yield self.env.process(self.arrived_machine()) # Simulación de llegada a la máquina print('[{}] entra a la [{}] a lavarse a la hora {:,.2f}'.format(car,name_machine,self.env.now)) #Proceso de lavar el carro yield self.env.process(self.wash_car()) # Proceso del lavado del vehículo print('[{}] lavado en la [{}] a la hora {:,.2f}'.format(car,name_machine,self.env.now)) self.vacate_machine(name_machine) # Proceso de desocupar la máquina #Proceso de salida self.out_time = random.randint(2,5) # Tiempo de salida del carwash print('[{}] empieza a dirigirse a la salida del carwash a la hora {:,.2f}'.format(car,(self.env.now))) yield self.env.process(self.go_out_carwash()) # Proceso de salida del carwash print('[{}] sale del carwash a la hora {:,.2f}'.format(car,self.env.now)) class Simulation(): def __init__(self,init): self.init = init # Número de vehículos para iniciar la simulación self.name_car = 'vehiculo_{{}}' # Nombre del vehículo # Ejecuta la simulación def execute_simulation(self, env, machine, interval): car_wash = CarWash(env, machine)# Inicializa la lavandería con las variables de interes self.init_cars(env,car_wash) # Inicializamos los vehículos para la simulación # Mientras el tiempo de simulación no termina llegaran más carros yield env.timeout(random.randint(interval-2,interval+2)) # Tiempo de espera a que llegue un nuevo vehícu 10 self.init+=1 # Aumentamos id del vehículo a uno env.process(car_wash.arrived_car(self.name_car.format(self.init))) # Iniciamos proceso para lavar el veh ículo # Inicializa los primeros vehículos def init_cars(self,env,car_wash): for i in range(self.init): # Creamos vehículos hasta el valor inicial env.process(car_wash.arrived_car(self.name_car.format(i+1))) # Iniciamos proceso para lavar el vehículo Realizamos la creación de las nuevas variables de interes a partir de la refactorización realizada, obteniendo las siguientes variables. In [20]: NUM_CARS = 3 NUM MACHINE = 3 INTERVAL = 9 TIME SIM = 24Realizamos el proceso de simulación con el proceso refactorizado para ver los nuevos resultados In [21]: print('*'*10,'Simulación CarWash UPS','*'*10) random.seed(77)env = simpy.Environment() simulacion = Simulation(NUM_CARS) env.process(simulacion.execute_simulation(env, NUM_MACHINE, INTERVAL)) env.run(until = TIME_SIM) ****** Simulación CarWash UPS ******* [vehiculo_1] llega a la hora 0.00 [vehiculo_2] llega a la hora 0.00 [vehiculo_3] llega a la hora 0.00

[vehiculo_1] empieza a dirigirse a la [maquina_1] a la hora 0.00 [vehiculo_2] empieza a dirigirse a la [maquina_2] a la hora 0.00 [vehiculo_3] empieza a dirigirse a la [maquina_3] a la hora 0.00 [vehiculo_2] entra a la [maquina_2] a lavarse a la hora 2.00 [vehiculo_3] entra a la [maquina_3] a lavarse a la hora 2.00 [vehiculo_1] entra a la [maquina_1] a lavarse a la hora 3.00

[vehiculo_2] empieza a dirigirse a la salida del carwash a la hora 8.00

[vehiculo_1] empieza a dirigirse a la salida del carwash a la hora 8.00

[vehiculo_3] empieza a dirigirse a la salida del carwash a la hora 13.00

[vehiculo_4] empieza a dirigirse a la salida del carwash a la hora 23.00

[vehiculo_4] empieza a dirigirse a la [maquina_1] a la hora 12.00

[vehiculo_4] entra a la [maquina_1] a lavarse a la hora 17.00

[vehiculo_5] empieza a dirigirse a la [maquina_2] a la hora 20.00

[vehiculo_2] lavado en la [maquina_2] a la hora 8.00

[vehiculo_1] lavado en la [maquina_1] a la hora 8.00

[vehiculo_3] lavado en la [maquina_3] a la hora 13.00

[vehiculo_4] lavado en la [maquina_1] a la hora 23.00

[vehiculo 2] sale del carwash a la hora 12.00

[vehiculo_1] sale del carwash a la hora 13.00 [vehiculo_3] sale del carwash a la hora 16.00

[vehiculo_4] llega a la hora 9.00

[vehiculo_5] llega a la hora 20.00

Conclusiones

Mediante este trabajo, hemos aprendido a utilizar la librería Simpy y ademas a entender como funcionan los proceso de simulación en eventos discretos con la finalidad de apreciar de mejor manera la importancia de realizar una simulación para poder tomar decisiones a partir de ello. Ademas hemos aprendido el uso de nuevos comandos como yield que permite guardar los estados de un proceso y ejecutarlos en paralelo con otros.

Como podemos ver, con la refactorización podemos entender de mejor manera como se realiza el proceso de carwash para ciertos carros, ademas de ello

realizamos una programación ordenada y coherente con la finalidad de que se entienda con comantarios a la vez lo que se realiza