



**TEMA:**

Carwash Simulacion

**Nombre:** Esteban David Rosero Pérez

**Asignatura:**

Simulación

**Docente:**

Ing. Diego Quisi

**Fecha:**

Cuenca, 23 de Mayo de 2021

En el proyecto que vamos a realizar, nosotros vamos a simular como funciona un auto de lavados, para declarar la simulación utilizaremos la Librería Simpy y sus procesos.

Para entender el funcionamiento y simular el sistema real de un auto lavados debemos declarar algunos procesos que toma.

- Agregar y modificar el tiempo que se demora de transportar de la cola de espera a la máquina de lavado (1-3)
- Identificar el numero de la maquina utilizada dentro de cada proceso.
- Agregar y modificar el tiempo de salida de la maquina del negocio entre (2-6)
- Modificar el tiempo de lavado entre (5-11)

Primero debemos tomar en cuenta que para la simulación vamos a tener los siguientes parámetros de entrada:

Inicio: En este parámetro ingresamos el numero de carros que va a comenzar la simulación

Maquinas: en este parámetro colocamos el numero de maquinas que esta disponible para lavar

Intervalo: aquí colocamos el intervalo de tiempo que se demora cada carro para su lavado

Tiempo\_simulacion: en esta variable colocamos el tiempo de simulación que va a tener nuestro proyecto de simpy

```
if __name__ == '__main__':  
  
    inicio = 3  
    maquinas = 3  
    intervalo = 9  
    tiempo_simulacion = 24  
  
    env = simpy.Environment()  
    simulacion = Simulacion(inicio)  
    env.process(simulacion.ejecutar_simulacion(env, maquinas, intervalo))  
    env.run(until=tiempo_simulacion)
```

En este proyecto tenemos dos clases para la simulación que son las siguientes:

- Simulación
- carWash

En nuestra clase CarWash tenemos las funciones mas importantes de la simulación, nosotros aquí declaramos los puntos a tener en cuenta en la simulación, nosotros tenemos nuestro proceso padre es el cual abarca todos los procesos pequeños, por ejemplo nuestra clase padre es la llegada del vehículo a la lavadora en el cual nosotros tomamos en cuenta muchos otros procesos pequeños como que tiempo le toma la llegada a la maquina, el tiempo que le toma a la maquina lavar el vehículo y la salida del vehículo.

En nuestra clase padre nosotros vamos a tomar en cuenta que cada maquina que se usa esa máquina estará ocupada hasta terminar el proceso de lavado por cada vehículo,

Primero nosotros vamos a declarar un tiempo de llegada entre 1 a 3 minutos que le toma al carro llegar a la maquina, vamos a imprimir la hora en la que se traslado a la maquina y la hora en la que llego la maquina

Después nosotros vamos a declarar un tiempo de lavado que dependiendo el tipo de carro será un aleatorio de tiempo entre 5 minutos a 11 minutos, vamos a tomar en cuenta que solo son autos pequeños.

Después vamos a declarar el tiempo de salida en el cual el vehículo le toma entre el finalizado del lavado y el moverlo a su lugar inicial.

```
main.py > carWash > llegada_vehiculo
6 class carWash():
7     def __init__(self,entorno,maquinas):
8         self.env = entorno
9         self.maquinas = simpy.Resource(entorno, maquinas)
10
11     def llegada_maquina(self,tiempo):
12         yield self.env.timeout(tiempo)
13
14     def lavado_vehiculo(self,tiempo):
15         yield self.env.timeout(tiempo)
16
17     def salida_vehiculo(self,tiempo):
18         yield self.env.timeout(tiempo)
19
20     def llegada_vehiculo(self,vehiculo):
21         print('llega {} a la hora {:.2f}'.format(vehiculo, self.env.now))
22         with self.maquinas.request() as maquina:
23             yield maquina
24             #proceso de llegar a la maquina
25             tiempo_llegada = random.randint(1,3)
26             print('El {} se traslada a la maquina a la hora {:.2f}'.format(vehiculo,self.env.now))
27             yield self.env.process(self.llegada_maquina(tiempo_llegada))
28             print('el {} llego a la maquina a la hora {:.2f}'.format(vehiculo,self.env.now))
29             #proceso de lavado
30             tiempo_lavado = random.randint(5, 11)
31             yield self.env.process(self.lavado_vehiculo(tiempo_lavado))
32             print('El {} se acabo de lavar a la hora {:.2f}'.format(vehiculo,self.env.now))
33             #proceso de salida del vehiculo
34             tiempo_salida = random.randint(2, 6)
35             print('El {} se traslada a la salida {:.2f}'.format(vehiculo,self.env.now))
36             yield self.env.process(self.salida_vehiculo(tiempo_salida))
37             print('El {} salio a la hora {:.2f}'.format(vehiculo,self.env.now))
38
```

En nuestra clase Simulación tenemos dos funciones:

- ejecutar\_simulacion : Aquí realizamos la inicialización de la clase carwash en el cual como parámetro tiene el entorno y las maquinas que le declaramos al inicio. También inicializamos el proceso padre de la clase carWash. Dentro del proceso de simulación nosotros vamos a declarar que mientras sea true (La simulación este activa) prosiga con los procesos, después declaramos un tiempo que en nuestro caso con la variable intervalo tomamos un randomico de entre -2 y +2 en comparación con la variable intervalo, sumamos a la variable inicio y mandamos a correr nuestro proceso padre.
- Iniciar\_vehiculos: En este proceso nosotros llamamos a la clase padre pero con un rango de carros iniciales en nuestro caso 3.

```

class Simulacion():

    def __init__(self, inicio):
        self.inicio = inicio
        self.nombre_vehiculo = 'vehiculo_{'

    def ejecutar_simulacion(self, env, maquinas, intervalo):
        car_wash = carWash(env, maquinas)
        self.iniciar_vehiculos(env, car_wash)

        while True:
            yield env.timeout(random.randint(intervalo-2, intervalo+2))
            self.inicio+=1
            yield env.process(car_wash.llegada_vehiculo(self.nombre_vehiculo.format(self.inicio)))

    def iniciar_vehiculos(self, env, car_wash):
        for i in range(self.inicio):
            env.process(car_wash.llegada_vehiculo(self.nombre_vehiculo.format(i)))

```

Ahora solo nos falta declarar correr la simulación, con estas líneas de código inicializamos a la simulación, después de declarar las variables.

Primero inicializamos el entorno a ese entorno la mandamos el numero de autos que en este caso es 3 con la variable inicio, después mandamos a correr el proceso padre de la simulación llamado Ejecutar Simulación.

Por Ultimo, al entorno le mandamos a correr y le mandamos la variable tiempo de simulación, que esta indica a la simulación en que momento debe parar.

```

if __name__ == '__main__':

    inicio = 3
    maquinas = 3
    intervalo = 9
    tiempo_simulacion = 24

    env = simpy.Environment()
    simulacion = Simulacion(inicio)
    env.process(simulacion.ejecutar_simulacion(env, maquinas, intervalo))
    env.run(until=tiempo_simulacion)

```

### Pruebas de la Practica:

Como podemos observar nosotros habíamos declarado que vamos a comenzar con 3 autos y 3 máquinas, por lo que llegan 3 autos al mismo tiempo y se comienzan a trasladar al mismo tiempo a sus respectivas maquinas.

Después como declaramos que tendría un aleatorio de tiempo de llegada entre el puesto a la maquina variaba entre 1 a 3 minutos:

El vehículo 2 llega primero se demoró dos minutos

El vehículo 0 y vehículo 1 se demoró 3 minutos para llegar a la maquina

Podemos observar que viene otro carro a los 8 minutos, y el vehículo 0 como también el vehículo 1 se terminaron de lavar a las 8, inmediatamente se le trasladaron a la salida, pero el vehículo numero 1 salió de la lavadora a los 10 minutos mientras que el vehículo que también se termino de lavar al mismo tiempo salió a los 11 minutos.

Como ya tenemos dos máquinas libres el vehículo numero 4 se traslada a la maquina a los 10 minutos.

El vehículo 2 se acaba de lavar al minuto 13 y se traslada inmediatamente a la salida, pero salió de la lavadora al minuto 18.

El ultimo vehículo que ingreso se acabo de lavar a los 21 minutos y se le llevo a la salida inmediatamente, como pusimos un tiempo limite de 24 segundos la simulación se termino y no nos indico a que hora salió el vehículo 4 de la lavadora.

```
(base) C:\Users\EstebanRM\Desktop\Esteban\universidad\10 semestre\simulacion\simpy>python main.py
llega vehiculo_0 a la hora 0.00
llega vehiculo_1 a la hora 0.00
llega vehiculo_2 a la hora 0.00
El vehiculo_0 seb traslada a la maquina a la hora 0.00
El vehiculo_1 seb traslada a la maquina a la hora 0.00
El vehiculo_2 seb traslada a la maquina a la hora 0.00
el vehiculo_2 llego a la maquina a la hora 2.00
el vehiculo_0 llego a la maquina a la hora 3.00
el vehiculo_1 llego a la maquina a la hora 3.00
llega vehiculo_4 a la hora 8.00
El vehiculo_0 se acabo de lavar a la hora 8.00
El vehiculo_0 se traslada a la salida 8.00
El vehiculo_1 se acabo de lavar a la hora 8.00
El vehiculo_1 se traslada a la salida 8.00
El vehiculo_1 salio a la hora 10.00
El vehiculo_4 seb traslada a la maquina a la hora 10.00
El vehiculo_0 salio a la hora 11.00
el vehiculo_4 llego a la maquina a la hora 11.00
El vehiculo_2 se acabo de lavar a la hora 13.00
El vehiculo_2 se traslada a la salida 13.00
El vehiculo_2 salio a la hora 18.00
El vehiculo_4 se acabo de lavar a la hora 21.00
El vehiculo_4 se traslada a la salida 21.00
```

## Conclusiones:

La librería simpy nos ayuda en este tipo de simulaciones, para poder entender como nuestro flujo de la vida real puede pasar con variables que se tomen con dichos datos, llevándolos a la simulación, buscando respuestas con diferentes factores de diferencia en cada variable.

La librería en este caso nos ayudo a simplificar la programación al momento de la ocupación de las maquinas en el cual solo un auto podía ocupar una maquina y los otros autos pasaban a la espera.