.UBAfiuba
FACULTAD DE INGENIERÍA

SimPy CheatSheet

Comandos básicos

Instala la biblioteca

```
!pip install simpy
```

Importa la librería SimPy

```
import simpy as sim
```

Crea el ambiente de simulación

```
env = sim.Environment()
```

Vincula una función al ambiente de simulación

```
env.process(función())
```

Inicia la simulación hasta el tiempo t

```
env.run(until=t)
```

Números aleatorios (numpy)

Devuelve alguna de las n opciones en forma aleatoria

```
np.random.choice(['F','M'], p=[0.5, 0.5])
```

Genera valores según una distribución normal

```
np.random.normal(media, desvío)
```

Genera valores según una distribución Gama

```
np.random.gamma(shape, scale=1.0)
```

Genera valores según una distribución uniforme

```
np.random.uniform(min, max)
```

Recursos

Alocación de recursos (creación del pool)

```
A = sim.Resource(env, q)
```

Solicitud de recurso

```
r = A.request()
```

Liberación de recurso

```
A.release(r)
```

Estado de una solicitud

```
r.triggered
```

Cantidad de recursos utilizados

```
A.count
```

Tiempo

Retorna el tiempo de la simulación

```
env.now
```

Genera una demora de tiempo t

```
yield env.timeout(t)
```

*Generación de un ambiente de tiempo real **

```
env = sim.RealtimeEnvironment(factor=1)
```

*Parar un proceso (en un generador)**

```
return None
```

Almacenes

Creación del almacén

```
S = sim.Store(env, Qmax)
```

Agrega un ítem al almacén

```
S.put(item)
```

Saca un ítem del almacén (FIFO)

```
S.get()
```

Lista los ítems del almacén

```
S.items
```

Retorna la capacidad del almacén (Qmax)

```
S.capacity
```

Contenedores

Creación del contenedor

```
T = sim.Container(env, Qmax)
```

Agrega contenido

```
T.put(Q)
```

Consume contenido

```
T.get(Q)
```

Retorna el nivel del contenedor

```
T.level
```