

Simulación Comedor Universitario

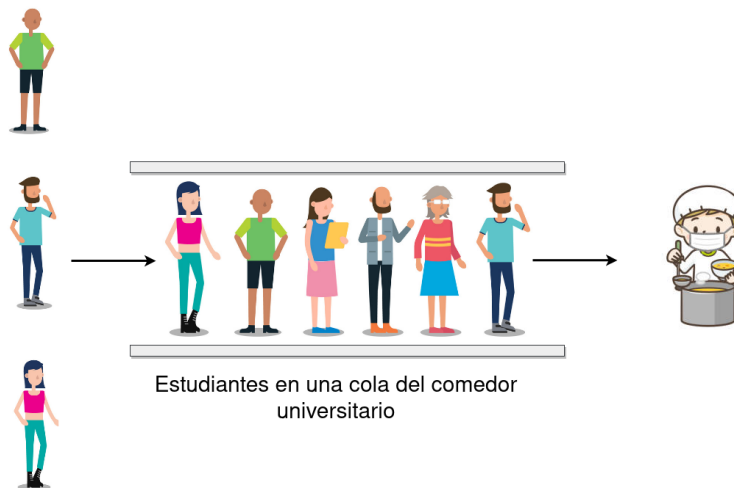
La programación se convierte en una herramienta importante para la construcción de simulaciones para predecir el comportamiento de las medidas de desempeño de un servicio, con el fin de obtener de manera oportuna información que permita encontrar un equilibrio entre la prestación de un servicio y la satisfacción de los clientes. En este caso nos apoyaremos de las Colas que nos permite crear simulaciones de situaciones cotidianas donde involucre una fila de clientes para ser atendidos en un servicio.

Una simulación más interesante nos permite estudiar el comportamiento de la atención del comedor universitario. Recuerde que a medida que los estudiantes llegan para recibir su almuerzo, estos se ubican en una cola para esperar ser atendidos según su orden de llegada. Muchas preguntas surgen, la más importante es si el comedor universitario es capaz de manejar una cierta cantidad de trabajo. Si no puede, los estudiantes estarán esperando demasiado tiempo para almorzar y pueden perder la próxima clase.

Considere la siguiente situación en el comedor universitario. En cualquier día promedio, alrededor de **800** estudiantes necesitan almorzar en cualquier momento durante la operación del servicio. Los funcionarios que despachan los almuerzos son capaces de procesar en promedio 1 estudiante por μ segundos, esto lo llamaremos tasa de servicio.

Podríamos tomar una decisión al respecto mediante la construcción de una simulación que modele el comedor universitario. Necesitamos construir representaciones para los estudiantes y comedor. A medida que los estudiantes llegan para almorzar, los agregamos a una lista de espera, una cola de estudiantes antes del comedor (momento que escanean el carnet). Cuando el comedor despacha a un estudiante, examinará la cola para ver si hay otros estudiantes pendientes de recibir el almuerzo. Es de interés para nosotros la cantidad promedio de tiempo que los estudiantes esperarán a que sus almuerzos sean servidos. Ésta es igual al promedio de tiempo que un estudiante espera en la cola.

Estudiantes



Para modelar esta situación necesitamos usar algunas probabilidades. Por ejemplo, si llegan λ estudiantes por hora. ¿Cuál es la probabilidad de que, en un segundo determinado, vaya a ingresar un estudiante a la cola del comedor? El modo para responder a esto es considerar la proporción entre estudiantes y tiempo. Si hay λ estudiantes por hora significa que en promedio habrá un estudiante cada S segundos.

$$\frac{\lambda \text{ estudiantes}}{1 \text{ hora}} * \frac{1 \text{ hora}}{60 \text{ minutos}} * \frac{1 \text{ minuto}}{60 \text{ segundos}} = \frac{1 \text{ estudiante}}{S \text{ segundos}}$$

Por cada segundo podemos simular la posibilidad de que se produzca una tarea de impresión generando un número aleatorio entre 1 y S inclusive. Si el número es S , decimos que un estudiante ha llegado a la fila. Tenga en cuenta que es posible que muchos estudiantes lleguen uno tras otro o quizás debamos esperar un buen rato para que aparezca un estudiante. Ésa es la naturaleza de la simulación. Usted desea simular la situación real de la manera más cercana posible dado que conoce los parámetros generales.

Pasos para la simulación

Aquí está la simulación principal.

1. Crear una cola de estudiantes en el comedor universitario. A cada estudiante se le dará una marca de tiempo (`marcaTiempo`) a su llegada. La cola está vacía al comenzar.
2. Para cada segundo (`segundoActual`):
 - a. ¿Llegó un nuevo estudiante? Si es así, agregarla a la cola con `segundoActual` como la marca de tiempo.
 - b. Si el comedor no está ocupado y hay estudiantes esperando,
 - i. Extraer el siguiente estudiante de la cola y asignarlo al comedor (el servicio de despacho de almuerzos).
 - ii. Restar `marcaTiempo` de `segundoActual` para calcular el tiempo de espera para ese estudiante.
 - iii. Añadir el tiempo de espera para ese estudiante a una lista para su procesamiento posterior.
 - c. Cuando el comedor atiende un estudiante resta un segundo del tiempo requerido para ese estudiante.
 - d. Si el comedor despacha a un estudiante, en otras palabras, el tiempo requerido ha llegado a cero, el comedor ya no está ocupado.
3. Una vez completada la simulación, calcule el tiempo de espera promedio usando la lista de tiempos de espera generados.

Implementación en Pseudocódigo

Para diseñar esta simulación crearemos clases para los tres objetos del mundo real descritos anteriormente: **Comedor**, **Estudiante** y **colaComedor**.

La clase Comedor (ver programa 1) tendrá que verificar si tiene un estudiante atendiendo. Si así es, entonces está ocupado (líneas 13-17) y la cantidad de tiempo necesario es la tasa de servicio μ . El constructor también permitirá que se inicialice la característica de tasa de servicios. El método **tictac** disminuye el temporizador interno y pone el comedor disponible (línea 11) si la tarea se ha completado.

Programa 1

```
1 class Comedor:
2     Constructor(servicio):
3         tasaServicio = servicio
4         estudianteActual = None
5         tiempoRestante = 0
6
7     def tictac():
8         if estudianteActual != None:
9             tiempoRestante = tiempoRestante - 1
10            if tiempoRestante == 0:
11                estudianteActual = None
12
13    def ocupada():
14        if estudianteActual != None:
15            return True
16        else:
17            return False
18
19    def iniciarNueva(nuevoEstudiante):
20        estudianteActual = nuevoEstudiante
21        tiempoRestante = tasaServicio
```

La clase Estudiante (ver el Programa 2) representará un solo estudiante. Cada estudiante también tendrá que mantener una marca de tiempo que se utilizará para calcular el tiempo de espera. Esta marca de tiempo representará la hora en que se creó el estudiante y se colocó en la cola. El método **tiempoEspera** puede utilizarse para obtener la cantidad de tiempo que el estudiante pasó en la cola antes de que comenzaran a servirle el almuerzo.

Programa 2

```

1  class Estudiante:
2      def Constructor(tiempo):
3          marcaTiempo = tiempo
4
5      def obtenerMarca():
6          return marcaTiempo
7
8      def tiempoEspera(tiempoActual):
9          return tiempoActual - marcaTiempo
    
```

La simulación principal (ver el Programa 3) implementa el algoritmo descrito anteriormente. El objeto `colaComedor` es una instancia de nuestro TAD Queue existente. Una función auxiliar booleana, `nuevoEstudiante`, decide si se ha llegado un nuevo estudiante. Los estudiantes aparecen una vez cada **S** segundos. Podemos simular este evento aleatorio eligiendo arbitrariamente **S** de entre el rango de enteros aleatorios (línea 28). La función de simulación nos permite ajustar el tiempo total y la tasa de servicio del comedor.

Programa 3

```

1  def simulacion(numeroSegundos, tasaServicio):
2
3      comedorUniversitario = Comedor(tasaServicio)
4      colaComedor = Queue()
5      tiemposEspera = List()
6
7      for segundoActual in range(numeroSegundos):
8
9          if nuevoEstudiante():
10             estudiante = Estudiante(segundoActual)
11             colaComedor.agregar(estudiante)
12
13             if (not comedorUniversitario.ocupada()) and (colaComedor.size() > 0):
14                 estudianteSiguiente = colaComedor.top()
15                 colaComedor.pop()
16                 tiemposEspera.insert(tiemposEspera.size(), estudianteSiguiente.tiempoEspera(segundoActual))
17                 comedorUniversitario.iniciarNueva(estudianteSiguiente)
18
19             comedorUniversitario.tictac()
20
21     esperaPromedio=sum(tiemposEspera)/len(tiemposEspera)
22     print("Tiempo de espera promedio%6.2f segundos %3d estudiantes
    
```

```
23 restantes."%(esperaPromedio, colaComedor.size()))
24
25
26 def nuevoEstudiante():
27     numero = random(1,S)
28     if numero == S:
29         return True
30     else:
31         return False
32
33
34 # 100 = cantidad de experimentos que vamos a correr la simulación
35 # 3600 es la cantidad de segundos en una hora
36 for i in range(100):
    simulacion(3600,5)
```

Diseño del experimento

Cuando ejecutamos la simulación, no debemos preocuparnos de que los resultados sean diferentes cada vez. Esto se debe a la naturaleza probabilística de los números aleatorios. Estamos interesados en las tendencias que pueden estar ocurriendo a medida que se ajustan los parámetros de la simulación.

En primer lugar, ejecutaremos la simulación durante un periodo de 60 minutos (3600 segundos) utilizando una tasa de servicio que deben calcular en sitio. Además, ejecutaremos 100 ejecuciones independientes. Recuerde que debido a que la simulación funciona con números aleatorios, cada ejecución retornará resultados diferentes. Para esto deben correr la simulación 100 veces y sacar un promedio de los tiempos de espera promedio.

Ejercicios

1. Calcular la tasa de llegada de estudiantes por hora. **λ** . Demuestre su respuesta.
 - a. Deben sacar un promedio de llegada visitando el sitio y contando los estudiantes que llegan durante una hora. Esto deben realizarlo en un periodo de 4 días. Al final sacan un promedio de esos días.
2. Calcular la tasa de servicio promedio. **μ** . Es decir, el tiempo que dura un estudiante desde que escanean el carnet hasta que sale de la cocina con su bandeja de comida. Demuestre su respuesta.
 - a. Deben sacar un promedio de servicio visitando el sitio y contando los segundos que un estudiante demora en que le despachen el almuerzo una vez ingresado al módulo de escaneo de carnet. Esto deben realizarlo en un periodo de 3 días. Al final sacan un promedio de esos días.

3. ¿Cuál es la cantidad de promedio de tiempo que los estudiantes deben esperar para que los atiendan?. Justifique su respuesta.
4. ¿En promedio cuántos estudiantes faltan por atender pasada la hora de servicio de la simulación?. Justifique su respuesta.
5. ¿Cuál debe ser la tasa de servicio si queremos que los estudiantes esperen en promedio 30 minutos en la cola?. Justifique su respuesta.
6. ¿Cómo modificaría la simulación para reflejar un mayor número de estudiantes? Supongamos que el próximo semestre duplicaremos la cantidad de estudiantes que reciben la ayuda del almuerzo. Usted tendría que hacer algunas suposiciones razonables sobre cómo se compuso esta simulación pero ¿qué cambiaría? Modifique el código.
7. Continuando con la pregunta anterior sobre duplicar la cantidad de estudiantes, ¿cuánto debería ser el promedio de tiempo de servicio para que el promedio del tiempo de espera por estudiante sea de 30 minutos?. Justifique su respuesta.
8. Por último ¿Cómo parametrizar el número de estudiantes?, en lugar de cambiar el código nos gustaría que el número de estudiantes sea un parámetro de la simulación.