# Proyecto de Simulación basada en Eventos Discretos

Dayany Alfaro González-C411

## 1. Problema: La Cocina de Kojo (Kojo's Kitchen)

La cocina de Kojo es uno de los puestos de comida rápida en un centro comercial. El centro comercial está abierto entre las 10:00 am y las 9:00 pm cada día. En este lugar se sirven dos tipos de productos: sándwiches y sushi. Para los objetivos de este proyecto se asumirá que existen solo dos tipos de consumidores: unos consumen solo sándwiches y los otros consumen solo productos de la gama del sushi. En Kojo hay dos períodos de hora pico durante un día de trabajo; uno entre las 11:30 am y la 1:30 pm, y el otro entre las 5:00 pm y las 7:00 pm. El intervalo de tiempo entre el arribo de un consumidor y el de otro no es homogéneo pero, por conveniencia, se asumirá que es homogéneo. El intervalo de tiempo de los segmentos homogéneos, distribuye de forma exponencial.

Actualmente dos empleados trabajan todo el día preparando sándwiches y sushi para los consumidores. El tiempo de preparación depende del producto en cuestión. Estos distribuyen de forma uniforme, en un rango de 3 a 5 minutos para la preparación de sándwiches y entre 5 y 8 minutos para la preparación de sushi.

El administrador de Kojo está muy feliz con el negocio, pero ha estado recibiendo quejas de los consumidores por la demora de sus peticiones. Él está interesado en explorar algunas opciones de distribución del personal para reducir el número de quejas. Su interés está centrado en comparar la situación actual con una opción alternativa donde se emplea un tercer empleado durante los períodos más ocupados. La medida del desempeño de estas opciones estará dada por el porciento de consumidores que espera más de 5 minutos por un servicio durante el curso de un día de trabajo.

Se desea obtener el porciento de consumidores que esperan más de 5 minutos cuando solo dos empleados están trabajando y este mismo dato agregando un empleado en las horas pico.

## 2. Principales ideas seguidas para la solución

Para resolver el problema se debe simular la llegada de personas al local, el cual abre desde un tiempo $t_0$ a un tiempo $T$, las personas pueden ser de 2 tipos, las que ordenan sushi y las que ordenan sándwich, la diferencia entre ellas es el tiempo que se demora el trabajador en procesarlas. Para determinar la orden de cada consumidor se decidió hacer uso de una variable aleatoria que distribuye Bernoulli de parámetro $p = 1/2$ definida de la siguiente forma:

$$\text{Orden}(u) = \begin{cases} \textit{sándwich} & u <= 1/2 \\ \textit{sushi} & u > 1/2 \end{cases}$$

Además tiene q $u$ distribuye uniforme en $[0, 1]$ de modo que existe la misma probabilidad de que un cliente pida sushi o sándwich.

Es necesario resolver 2 variantes del problema y compararlas:

1. Simular el problema con 2 trabajadores.
2. Simular el problema con 2 trabajadores y un trabajador adicional en horarios pico.

Para llegar a la solución del problema es fundamental percatarse de que la segunda variante contiene a la primera. Por tanto, si se sabe cómo resolver la primera variante, solo se necesita agregar el comportamiento extra en horarios pico para la segunda variante.

## 3. Modelo de Simulación de Eventos Discretos Desarrollado para resolver el problema

Inicialmente se generó una implementación del modelo **Sistema de atención con $n$ servidores en paralelo**, esto permite resolver la primera versión del problema cuando $n = 2$.

Lista de variables llevadas durante la simulación:

- Variable de tiempo
  - **elapsed_time**
- Variables contadoras
  - **number_arrives**: cantidad de llegadas.
  - **attended_by_worker**$_j$: cantidad de clientes atendidos por el trabajador $j$, $j \in \{1, 2.., n\}$.
- Variables de estado del sistema
  - **number_customers**: cantidad de clientes actualmente en el sistema.
  - **customers_to_attend**: cola de clientes que están esperando para ser atendidos.

- **workers**$_j$: id del cliente al que está atendiendo el trabajador $j$, $j \in \{1, 2.., n\}$. Si $j$ no está atendiendo a ningún cliente workers$_j = 0$

- Variables de Salida
  - **customers**: contiene para cada cliente que fue procesado los siguientes datos
    - *id*: número asignado al llegar al sistema
    - *order*: 0 si el cliente ordena sushi, 1 si ordena sandwich
    - *arrived*: tiempo en que llegó el cliente.
    - *attended*: tiempo en que un trabajador comenzó a procesar su pedido.
    - *departure*: tiempo cuando su pedido fue culminado y se le entregó al cliente, el cual coincide con el tiempo en que el cliente se va del local.

- Lista de eventos
  - ($t_a$, **service_time**$_j$): $t_a$ es el tiempo del próximo arribo y service_time$_j$ es el tiempo de partida del cliente que está siendo atendido por el trabajador $j$, $j \in \{1, 2.., n\}$. Si $j$ no está atendiendo a ningún cliente service_time$_j = \infty$

- Inicialización

  elapsed_time = number_arrives = number_customers = 0
  attended_by_worker$_j = 0 \quad j \in \{1, 2.., n\}$
  workers$_j = 0 \quad j \in \{1, 2.., n\}$
  service_time$_j = \infty \quad j \in \{1, 2.., n\}$
  Generar $T_0$
  $t_a = T_0$

- Caso 1: $t_a = min(t_a$, service_time$_j)$

  elapsed_time = $t_a$
  number_arrives += 1
  number_customers += 1
  customer.id = number_arrives
  customer.order = Generar orden
  customer.arrive = $t_a$
  Generar $T_t$
  $t_a = T_t$
  Si $\exists$ workers$_i = 0$:
  {
  workers$_i$ = customer.id
  customer.attended = elapsed_time
  attended_by_worker$_i$ += 1
  Generar $Y$ según orden
  service_time$_i$ = elapsed_time + $Y$
  }

Si $\text{workers}_i \neq 0 \ \forall i,\ i \in \{1, 2.., n\}$
{
customers_to_attend = [... ,customer.id]
}

- Caso 2: $t_i = min(t_a, \text{service\_time}_j)$

  elapsed_time = $t_i$
  customer.departure = $t_i$
  number_customers -= 1
  Si customers_to_attend = [customer,...]:
  {
  $\text{workers}_i$ = customer.id
  customer.attended = $t_i$
  $\text{attended\_by\_worker}_i$ += 1
  Generar $Y$ según orden
  $\text{service\_time}_i$ = elapsed_time + $Y$
  }
  Si customers_to_attend = [ ]:
  {
  $\text{workers}_i = 0$
  $\text{service\_time}_i = \infty$
  }

Para resolver el segundo problema se mantienen todos los elementos del primero y además se añaden los siguientes elementos.

- Variables Contadoras
  - **attended_by_worker**$_{n+1}$: cantidad de clientes atendidos por el trabajador adicional.
- Variables de estado del sistema
  - **workers**$_{n+1}$: cliente al que está atendiendo el trabajador adicional. Si no está atendiendo a ningún cliente $\text{workers}_{n+1} = 0$
- Eventos
  - **rush_hour_start**: comienzo de la hora pico, se activa el trabajador adicional.
  - **rush_hour_finish**: fin de la hora pico, se desactiva el trabajador adicional.
  - **service_time**$_{n+1}$: tiempo de partida del cliente que está siendo atendido por el trabajador adicional. Si no está atendiendo a ningún cliente $\text{service\_time}_j = \infty$
- Inicialización

attended_by_worker$_{n+1} = 0$
workers$_{n+1} = \infty$
service_time$_{n+1} = \infty$

- Caso 3: rush_hour_start

  workers$_{n+1} = 0$

- Caso 4: rush_hour_finish

  workers$_{n+1} = \infty$
  service_time$_{n+1} = \infty$

- Caso 5: service_time$_{n+1}$

  $t_i$ = service_time$_{n+1}$
  elapsed_time = $t_i$
  customer.departure = $t_i$
  number_customers -= 1
  Si customers_to_attend = [customer,...]:
  {
  workers$_{n+1}$ = customer.id
  customer.attended = $t_i$
  attended_by_worker$_{n+1}$ += 1
  Generar $Y$ según orden
  service_time$_{n+1}$ = elapsed_time + $Y$
  }
  Si customers_to_attend = []:
  {
  workers$_{n+1} = 0$
  service_time$_{n+1} = \infty$
  }

# 4. Consideraciones obtenidas a partir de la ejecución de las simulaciones del problema

Se realizó un total de 10000 simulaciones del problema, en cada una de sus variantes, para distintos valores del parámetro($\lambda$) de la exponencial que define los arribos de consumidores y se obtuvieron los resultados que se muestran a continuación en el Cuadro 1.

| $\lambda$ | 2 Trabajadores | 2 Trabajadores y Extra |
|---|---|---|
| 1/2 | 96.77 | 81.99 |
| 1/3 | 50.59 | 28.57 |
| 1/4 | 15.73 | 9.37 |
| 1/5 | 6.85 | 4.08 |
| 1/6 | 3.61 | 2.11 |
| 1/7 | 2.21 | 1.33 |
| 1/8 | 1.17 | 0.86 |

Cuadro 1: Porciento de personas que esperaron más de 5 minutos para ser atendidas.

Como se puede apreciar en el Cuadro 1 el porciento de personas que esperaron más de 5 minutos para ser atendidas por un trabajador ($consumer.attended - consumer.arrive$) es menor cuando se hace uso de un trabajador extra en el local en adición a los dos regulares. Los porcientos y las diferencias entre ellos van disminuyendo a medida que el valor de $\lambda$ disminuye, como es de esperar, pues aumenta el valor esperado del tiempo hasta que llegue el próximo cliente y es más probale que algún trabajador esté desocupado. Por tanto, especialmente en los casos en que la media del tiempo de llegada entre clientes es pequeña, la alternativa de usar un trabajador adicional presenta un desempeño superior.