

# Base model for queue theory

## Optimizados - Markov's Pizza

2024-12-04

### Descripción del Modelo Base

Markov's Pizza opera durante 16 horas diarias (desde las 12:00 PM hasta las 4:00 AM) con dos chefs trabajando simultáneamente para preparar pizzas para los clientes. Modelaremos la producción de pizzas como un sistema de colas para analizar la eficiencia y posibles cuellos de botella en el proceso de elaboración de pizzas.

### Características del Modelo de Colas

- **Tipo de Modelo:** Este escenario se ajusta a un Modelo de Cola Multi-Servidor (Cola M/M/2)
- **M (Llegada):** Distribución exponencial (Poisson) de las llegadas de los clientes
- **M (Servicio):** Distribución exponencial de los tiempos de preparación de las pizzas
- **2 (Servidores):** Dos chefs trabajando simultáneamente

### Parámetros Clave a Considerar

- **Tasa de Llegada ( $\lambda$ ):** Número promedio de pedidos de pizza por unidad de tiempo
- **Tasa de Servicio ( $\mu$ ):** Número promedio de pizzas que un solo chef puede preparar por unidad de tiempo, también tomando en cuenta el horno compartido entre los chefs
- **Capacidad del Sistema:** Número máximo de pedidos que pueden estar en el sistema (en cola y siendo preparados)
- **Disciplina de Cola:** Primero en Llegar, Primero en Ser Atendido (FCFS)

### Especificaciones del Horno

- Tamaño del horno: 56" horno comercial
- Capacidad: 10 pizzas simultáneamente
- Rendimiento teórico: 12 pizzas por posición por hora (5min por pizza)
- Rendimiento teórico total: 120 pizzas por hora
- Tiempo de horneado: 5 minutos por pizza

### Cálculo de la Tasa de Servicio del Chef y del Horno

- Tasa de preparación del chef:
  - Supongamos que cada chef tarda entre 5 y 7 minutos en preparar una pizza.
  - Promedio de 6 minutos por pizza.
  - Tasa de preparación por chef: 10 pizzas por hora (60min / 6min)
  - Tasa de preparación combinada de dos chefs: 20 pizzas por hora
- Tasa de servicio del horno:
  - Máximo teórico: 120 pizzas por hora
  - Rendimiento práctico: Supongamos una eficiencia del 80% (tiempo de calentamiento, cambiar fuente de gaz, etc.)
  - Tasa de servicio práctica del horno: 96 pizzas por hora

## Análisis

- Tiempo de espera promedio de un pedido
- Probabilidad de espera
- Ocio de los chefs
- “Ocio” del horno
- Longitud de la cola
- Punto de saturación del sistema
- Analizar posibles mejoras en el proceso tomando en cuenta los costos asociados (agregar más chefs, hornos, etc.)

## Dependencias del programa

- queueing

```
# Check if required packages are installed and install if not
if (!require("queueing")) {
  install.packages("queueing")
}
```

```
## Loading required package: queueing
```

```
if (!require("ggplot2")) {
  install.packages("ggplot2")
}
```

```
## Loading required package: ggplot2
```

```
if (!require("dplyr")) {
  install.packages("dplyr")
}
```

```
## Loading required package: dplyr
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
## filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## intersect, setdiff, setequal, union
```

```
# Load required libraries
```

```
library(queueing)
```

```
library(ggplot2)
```

```
library(dplyr)
```

```
# Set seed for reproducibility
```

```
set.seed(506)
```

## Definición de Parámetros

```
lambda <- 20 # Llegadas de pedidos por hora, dato hipotético
# TODO: ver cual dato real usar o como obtenerlo
```

```
arrival_rates <- data.frame(
```

```

period = c(
  "12-2 PM",
  "2-4 PM",
  "4-6 PM",
  "6-8 PM",
  "8-10 PM",
  "10 PM-12 AM",
  "12-2 AM",
  "2-4 AM"
),
start_hour = c(0, 2, 4, 6, 8, 10, 12, 14),
lambda = c(10, 25, 35, 50, 40, 20, 15, 5)
# pizzas perhours, mas consumo en la noche
)

mu_chef <- 10 # pizzas per hour for each chef
mu_oven <- 96 # pizzas per hour for the oven

# metadatos del sistema
num_chefs <- 2 # cantidad de chefs
system_capacity <- 50 # antidad máxima de pedidos en el sistema

total_hours <- 16 # total de horas por día
# TODO: verificar cuantos chefs trabajan en el turno de 16 horas
# suponemos 4 chefs (2 turnos de 8 horas)

```

## Cálculo del ocio y utilización de los chefs y del horno

```

# calculate metrics given a specific lambda,
# usefull now that we have different arrival rates
calculate_metrics <- function(lambda) {
  utilization_chefs <- lambda / (num_chefs * mu_chef)
  utilization_oven <- lambda / mu_oven

  idle_chefs <- 1 - utilization_chefs
  idle_oven <- 1 - utilization_oven

  list(
    lambda = lambda,
    utilization_chefs = utilization_chefs,
    utilization_oven = utilization_oven,
    idle_chefs = idle_chefs,
    idle_oven = idle_oven
  )
}

```

## Cáculamos las métricas para cada arrival rate

```

# calculate metrics for each arrival rate
metrics <- lapply(arrival_rates$lambda, calculate_metrics)

```

```

# # convert to data frame for plotting
# metrics_df <- do.call(rbind, metrics)
# metrics_df$period <- arrival_rates$period
# metrics_df$start_hour <- arrival_rates$start_hour
# no need to add lambda again, it's already in the data frame metrics

metrics_df <- data.frame(
  period = arrival_rates$period,
  start_hour = arrival_rates$start_hour,
  lambda = arrival_rates$lambda,
  utilization_chefs = sapply(metrics, `[`, "utilization_chefs"),
  utilization_oven = sapply(metrics, `[`, "utilization_oven"),
  idle_chefs = sapply(metrics, `[`, "idle_chefs"),
  idle_oven = sapply(metrics, `[`, "idle_oven")
)

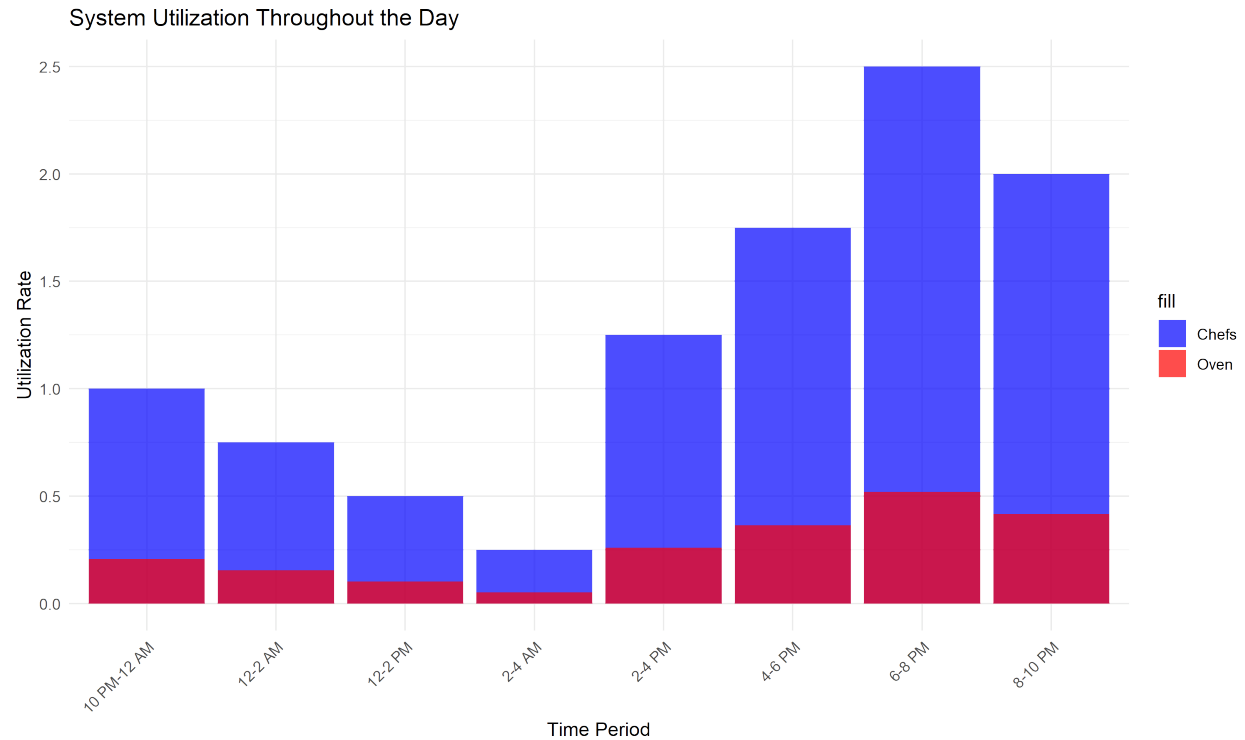
```

## visualizamos las métricas

```

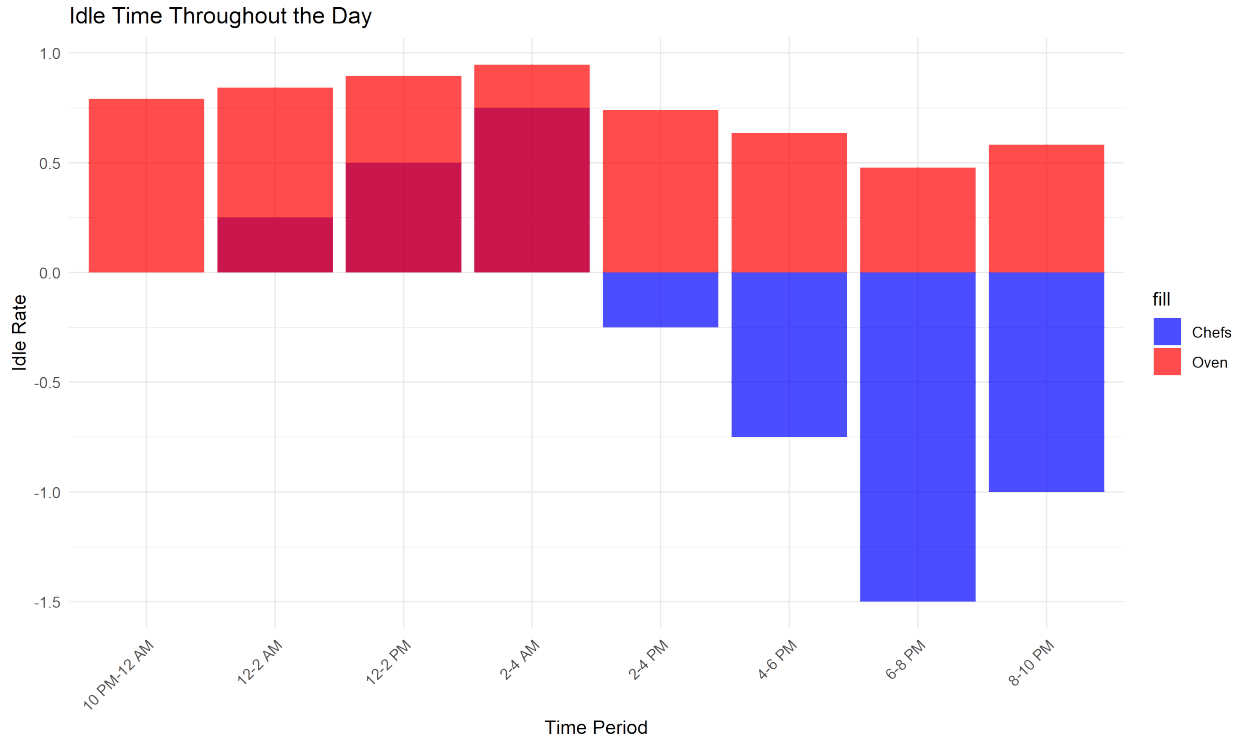
# Plot utilization metrics
ggplot(metrics_df, aes(x = period)) +
  geom_bar(aes(y = utilization_chefs, fill = "Chefs"),
    stat = "identity", position = "dodge", alpha = 0.7) +
  geom_bar(aes(y = utilization_oven, fill = "Oven"),
    stat = "identity", position = "dodge", alpha = 0.7) +
  labs(title = "System Utilization Throughout the Day",
    x = "Time Period",
    y = "Utilization Rate") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("Chefs" = "blue", "Oven" = "red"))

```



```
# store the plot as image
ggsave("utilization_metrics.png", width = 10, height = 6, dpi = 300)

# Plot idle time metrics
ggplot(metrics_df, aes(x = period)) +
  geom_bar(aes(y = idle_chefs, fill = "Chefs"),
    stat = "identity", position = "dodge", alpha = 0.7) +
  geom_bar(aes(y = idle_oven, fill = "Oven"),
    stat = "identity", position = "dodge", alpha = 0.7) +
  labs(title = "Idle Time Throughout the Day",
    x = "Time Period",
    y = "Idle Rate") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("Chefs" = "blue", "Oven" = "red"))
```



```
# store the plots as images
ggsave("idle_metrics.png", width = 10, height = 6, dpi = 300)

# Print the metrics dataframe for additional insight
print(metrics_df)
```

```
##      period start_hour lambda utilization_chefs utilization_oven idle_chefs
## 1 12-2 PM          0     10           0.50         0.10416667      0.50
## 2  2-4 PM          2     25           1.25         0.26041667     -0.25
## 3  4-6 PM          4     35           1.75         0.36458333     -0.75
## 4  6-8 PM          6     50           2.50         0.52083333     -1.50
## 5  8-10 PM         8     40           2.00         0.41666667     -1.00
## 6 10 PM-12 AM      10     20           1.00         0.20833333      0.00
## 7 12-2 AM         12     15           0.75         0.15625000      0.25
## 8  2-4 AM         14      5           0.25         0.05208333      0.75
##      idle_oven
## 1 0.8958333
## 2 0.7395833
## 3 0.6354167
## 4 0.4791667
## 5 0.5833333
## 6 0.7916667
## 7 0.8437500
## 8 0.9479167
```

## Analisis de los resultados iniciales

Notamos que al tener 2 chefs trabajando simultáneamente, es suficiente para los segmentos del día de 10pm a 4am, pero entre 2pm a 10pm no son suficientes. el horno nunca sobrepasa el 50% de utilización, por lo que no es un cuello de botella. Los datos teoricos encontrados presentan una venta de 200 pizzas por día, al

estudiar nuestro sistema inicial encontramos que es imposible llegar a esa cantidad con 2 chefs, para optimizar ganancias y costos se presentaran modelos alternativos para mejorar el sistema.