
TP 3 - SIMULACIÓN DE UN MODELO MM1 E INVENTARIO

Lautaro Teta Musa
Universidad Tecnológica Nacional
lautarotetamusa@gmail.com

Ignacio Roca
Universidad Tecnológica Nacional
ignacioroca@gmail.com

Agustín Luzzini
Universidad Tecnológica Nacional
agusluzzini@gmail.com

Nazareno Necchi
Universidad Tecnológica Nacional
nazanecchi.cer@gmail.com

7 de agosto, 2024

ABSTRACT

Este informe presenta una simulación de los modelos MM1 y de inventario (Q, R). Para el modelo MM1, se analizan métricas como el promedio de clientes en el sistema y en cola, tiempo promedio, utilización del servidor, y probabilidades de clientes en cola y denegación de servicio. Se varían las tasas de arribo en relación con la tasa de servicio en 25%, 50%, 75%, 100%, y 125%, realizando 10 corridas por experimento. En el modelo de inventario, se evalúan costos de orden, mantenimiento, faltante y total. Se comparan resultados teóricos, simulaciones en Python y una implementación en AnyLogic, presentando gráficas y análisis. Además, se desarrolla una interfaz interactiva para la entrada de parámetros y visualización de resultados en tiempo real.

Keywords Simulación · Modelo MM1 · Modelo de Inventario · Teoría de Colas · Simpy · AnyLogic · Análisis Comparativo · Parámetros de Entrada.

1 Introducción

La optimización de sistemas de colas y de inventarios es fundamental para mejorar la eficiencia operativa en múltiples contextos industriales y de servicios. Los modelos teóricos como el MM1 y el (Q, R) proporcionan herramientas valiosas para comprender y mejorar estos sistemas.

El modelo MM1, caracterizado por llegadas de clientes según un proceso de Poisson y tiempos de servicio exponenciales con un solo servidor, ofrece una visión simplificada pero poderosa de las dinámicas de espera y servicio. Este modelo es aplicable en una variedad de situaciones prácticas, como la atención al cliente, la gestión de tareas en sistemas informáticos y la administración de tráfico.

Por otro lado, los modelos de inventario, especialmente el modelo (Q, R), son esenciales para la gestión efectiva de stocks. En este modelo, la cantidad de pedido (Q) y el punto de reorden (R) se utilizan para equilibrar los costos de mantenimiento, orden y faltante, asegurando que los niveles de inventario sean óptimos y los costos operativos se mantengan bajos.

Este estudio se enfoca en un análisis detallado de las métricas de rendimiento de estos modelos. Para el modelo MM1, se examinan factores como el promedio de clientes en el sistema y en cola, el tiempo promedio en el sistema y en cola, la utilización del servidor y las probabilidades de clientes en cola y de denegación de servicio. Las tasas de arribo se variarán en relación con la tasa de servicio en múltiples proporciones para observar los efectos en el rendimiento del sistema.

En el caso del modelo de inventario, se evalúan los costos de orden, mantenimiento, faltante y el costo total, proporcionando una visión integral de cómo estos costos interactúan y afectan la eficiencia del sistema de inventario. Los resultados de este análisis se compararán con valores teóricos y se validarán mediante simulaciones en Python y una implementación en AnyLogic, ofreciendo una comprensión profunda y validada de los modelos.

2 Formulación teórica

2.1 Modelo de sistema de cola M/M/1

Un sistema de colas M/M/1 es un modelo en el que los clientes llegan según un proceso de Poisson y el servicio se presta de acuerdo a una distribución exponencial, con un único servidor. En este modelo:

- La tasa de llegada de los clientes se describe mediante la función:

$$\lambda(t) = \lambda e^{-\lambda t} \quad (1)$$

donde λ es la tasa de arribos, que indica el número promedio de llegadas por unidad de tiempo.

- La tasa de servicio sigue una distribución exponencial y está dada por:

$$\mu(t) = \mu e^{-\mu t} \quad (2)$$

donde μ es la tasa de servicio, representando el número promedio de clientes que el servidor puede atender por unidad de tiempo.

Calculamos algunos valores necesarios de la siguiente manera:

- Número promedio de clientes en el sistema (L_s):

$$L_s = \frac{\lambda}{\mu - \lambda} \quad (3)$$

- Número promedio de clientes en cola (L_q):

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} \quad (4)$$

- Tiempo promedio en el sistema (W_s):

$$W_s = \frac{1}{\mu - \lambda} \quad (5)$$

- Tiempo promedio en cola (W_q):

$$W_q = \frac{\lambda}{\mu(\mu - \lambda)} \quad (6)$$

- Utilización del servidor (p):

$$p = \frac{\lambda}{\mu} \quad (7)$$

- Probabilidad de que haya n clientes en el cola (P_{cola}):

$$P_{\text{cola}}(n) = (1 - \rho) \cdot \rho^{n+1} \quad (8)$$

- Probabilidad de denegación de servicio cuando la cola máxima es de l clientes (P_l):

$$P_{\text{denegación}} = P(l) = (1 - \rho) \cdot \rho^l \quad (9)$$

2.2 Modelo de inventario

En la evaluación del rendimiento de un modelo de inventario, es esencial cuantificar diferentes costos que afectan directamente la eficiencia y viabilidad del sistema. Estas medidas de rendimiento permiten analizar y comparar distintos escenarios, facilitando la toma de decisiones para optimizar el manejo del inventario. A continuación, se presentan las fórmulas para calcular los costos de orden, mantenimiento, faltante y el costo total, que son fundamentales para evaluar la efectividad del sistema de inventario en términos económicos. Calculamos algunos valores necesarios de la siguiente manera:

- Costo de Orden (C_o):

$$C_o = D \times S \quad (10)$$

El costo de orden se refiere al costo incurrido cada vez que se realiza un pedido, el cual se calcula multiplicando la demanda anual (D) por el costo de realizar un pedido (S).

- Costo de Mantenimiento (C_m):

$$C_m = \frac{Q}{2} \times H \quad (11)$$

El costo de mantenimiento representa el costo de mantener inventario en almacenamiento, calculado como el promedio del inventario ($Q/2$) multiplicado por el costo de mantener una unidad en inventario (H).

- Costo de Faltante (C_f):

$$C_f = \text{Costo por unidad faltante} \times \text{Número de unidades faltantes} \quad (12)$$

El costo de faltante se refiere a la penalización o pérdida de ingresos asociada con no tener suficiente inventario para cubrir la demanda.

- Costo Total (C_T):

$$C_T = C_o + C_m + C_f \quad (13)$$

El costo total es la suma de los costos de orden, mantenimiento y faltante, representando el costo global del sistema de inventario.

3 Desarrollo

3.1 Modelo de sistema de cola M/M/1

En esta sección, se analiza el comportamiento del sistema de colas M/M/1 con una tasa de arribo variable, correspondiente al : 25%, 50%, 75%, 100%, 125% con respecto a la tasa de servicio. Las métricas analizadas incluyen el número promedio de clientes en el sistema (L), el número promedio de clientes en cola (L_q), la utilización del servidor (ρ), el tiempo promedio en el sistema (W), el tiempo promedio en cola (W_q), la probabilidad de encontrar 3 clientes en cola y la probabilidad de denegación de servicio cuando la cola tiene una capacidad máxima de 10.

3.1.1 Tasa de arribo: 25%

Vamos a analizar el comportamiento del sistema de colas M/M/1 con una tasa de arribo correspondiente al : 25% de la tasa de servicio. Las métricas analizadas van a ser las mencionadas en la introducción, y se van a comparar a partir de 3 fuentes : la simulación en Python, los valores teóricos calculados y los resultados obtenidos con AnyLogic.

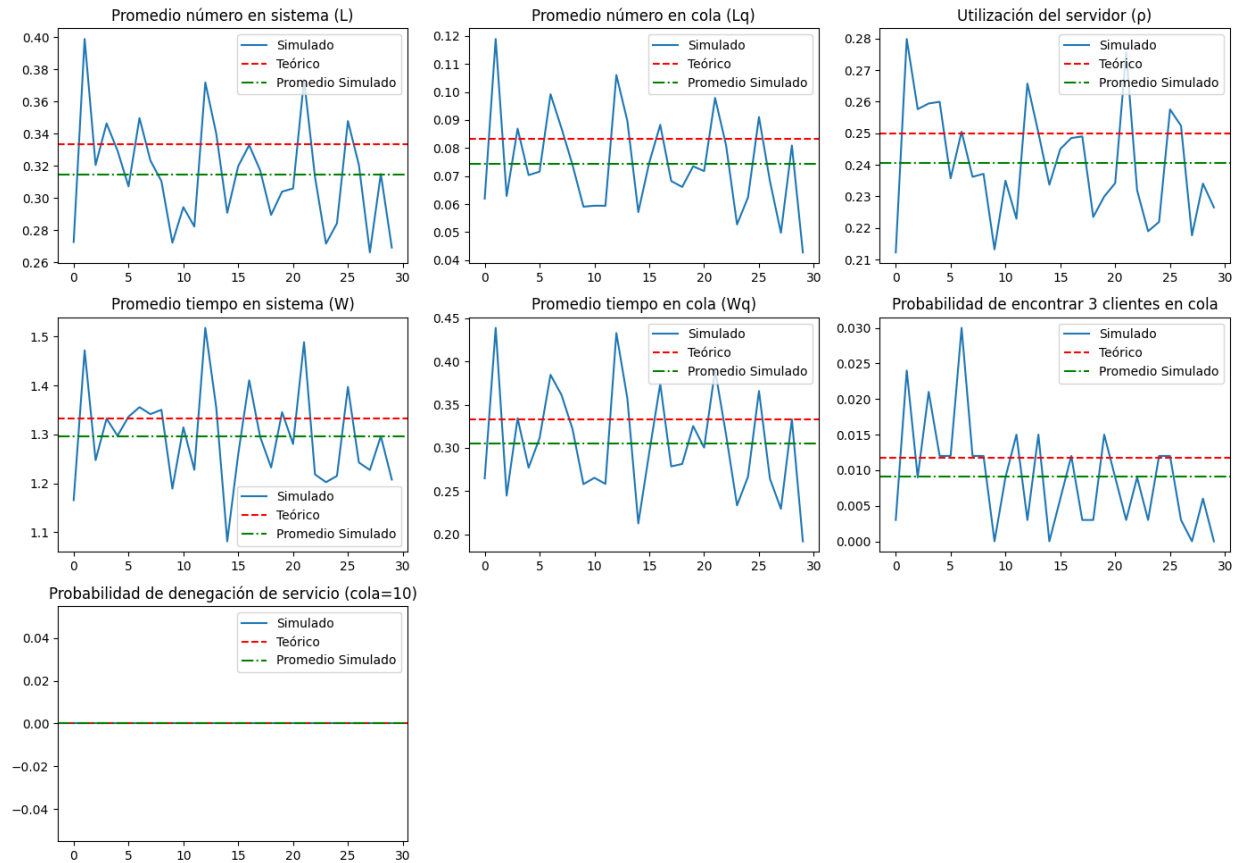


Figure 1: Comparación de resultados simulados y teóricos para una tasa de arribo del 25%.

En la Figura 1, se presentan los resultados obtenidos a partir de la simulación en Python y los valores teóricos calculados. Para el número promedio de clientes en el sistema (L), se observa que los valores simulados en Python muestran cierta variabilidad, oscilando alrededor del valor teórico esperado de 0.33.

El número promedio de clientes en cola (L_q) sigue un patrón similar, con los resultados simulados variando ligeramente en torno al valor teórico de 0.083.

La utilización del servidor (ρ) muestra una ligera fluctuación en la simulación, pero se mantiene cerca del valor teórico de 0.25.

El tiempo promedio en el sistema (W) y el tiempo promedio en cola (W_q) presentan variaciones en la simulación que oscilan alrededor de los valores teóricos de 1.33 y 0.33, respectivamente.

Por último, las probabilidades de encontrar 3 clientes en cola y de denegación de servicio también son consistentes entre la simulación y los valores teóricos, con pequeñas diferencias que pueden atribuirse a la naturaleza aleatoria del proceso de simulación.



Figure 2: Comparación de resultados de AnyLogic para una tasa de arribo del 25%.

En la Figura 2, se muestran los resultados obtenidos con AnyLogic luego de varias corridas eligiendo un escenario promedio.

Los resultados de AnyLogic muestran una alta coincidencia con los valores teóricos para todas las métricas analizadas. Las diferencias observadas entre los resultados simulados en Python y los obtenidos con AnyLogic son mínimas, lo que indica que ambos enfoques son válidos para modelar el sistema M/M/1 bajo una tasa de arribo del 25%.

En particular, AnyLogic presenta resultados ligeramente más estables, lo que puede deberse a las técnicas de simulación que utiliza, reduciendo la variabilidad observada en la simulación de Python.

Conclusión: Para una tasa de arribo del 25%, se observa una alta concordancia entre los resultados teóricos, simulados en Python y obtenidos con AnyLogic. Las pequeñas diferencias pueden deberse a la variabilidad inherente en las simulaciones, pero en general, todos los métodos confirman la validez del modelo M/M/1 bajo estas condiciones.

3.1.2 Tasa de arribo: 50%

Vamos a analizar el comportamiento del sistema de colas M/M/1 con una tasa de arribo correspondiente al : 50% de la tasa de servicio. Las métricas analizadas van a ser las mencionadas en la introduccion, y se van a comparar a partir de 3 fuentes : la simulación en Python, los valores teóricos calculados y los resultados obtenidos con AnyLogic.

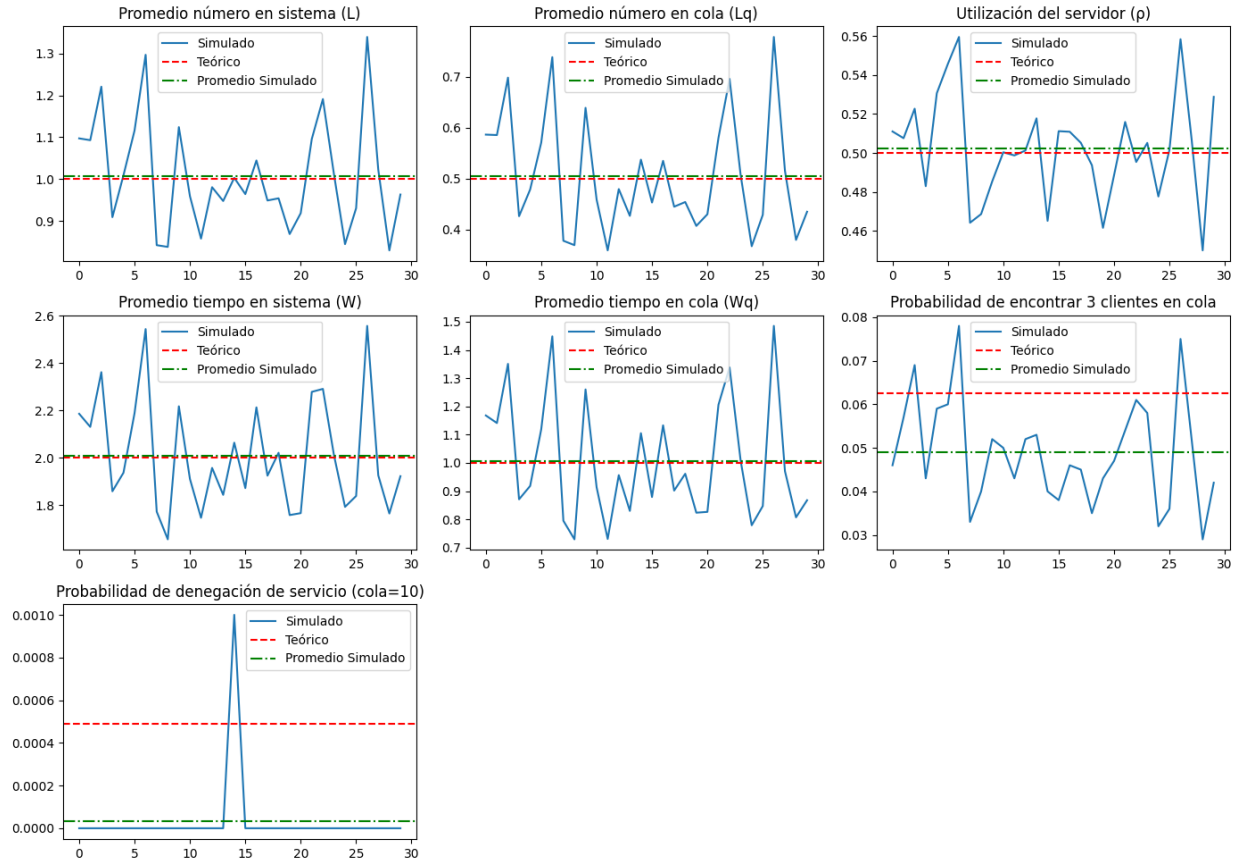


Figure 3: Comparación de resultados simulados en Python y valores teóricos para una tasa de arribo del 50%.

En la Figura 3, se presentan los resultados obtenidos a partir de la simulación en Python y los valores teóricos calculados.

Para el número promedio de clientes en el sistema (L), los valores simulados en Python muestran cierta variabilidad, oscilando alrededor del valor teórico esperado de 1.0. Este comportamiento es consistente con lo esperado para una tasa de arribo del 50%.

El número promedio de clientes en cola (L_q) muestra variaciones en los resultados simulados, que oscilan alrededor del valor teórico de 0.5. Esta variabilidad es común en simulaciones estocásticas, pero el promedio se alinea con los resultados teóricos.

La utilización del servidor (ρ) se mantiene cercana al valor teórico de 0.5, aunque con algunas fluctuaciones en los resultados simulados.

El tiempo promedio en el sistema (W) y el tiempo promedio en cola (W_q) presentan valores simulados que oscilan alrededor de 2.0 y 1.0, respectivamente, que son los valores teóricos esperados para una tasa de arribo del 50%.

Por último, la probabilidad de encontrar 3 clientes en cola y la probabilidad de denegación de servicio muestran fluctuaciones en los resultados simulados, pero en general, se alinean con los valores teóricos. La probabilidad de denegación de servicio, aunque pequeña, es perceptible en la simulación.

Comparación con AnyLogic:

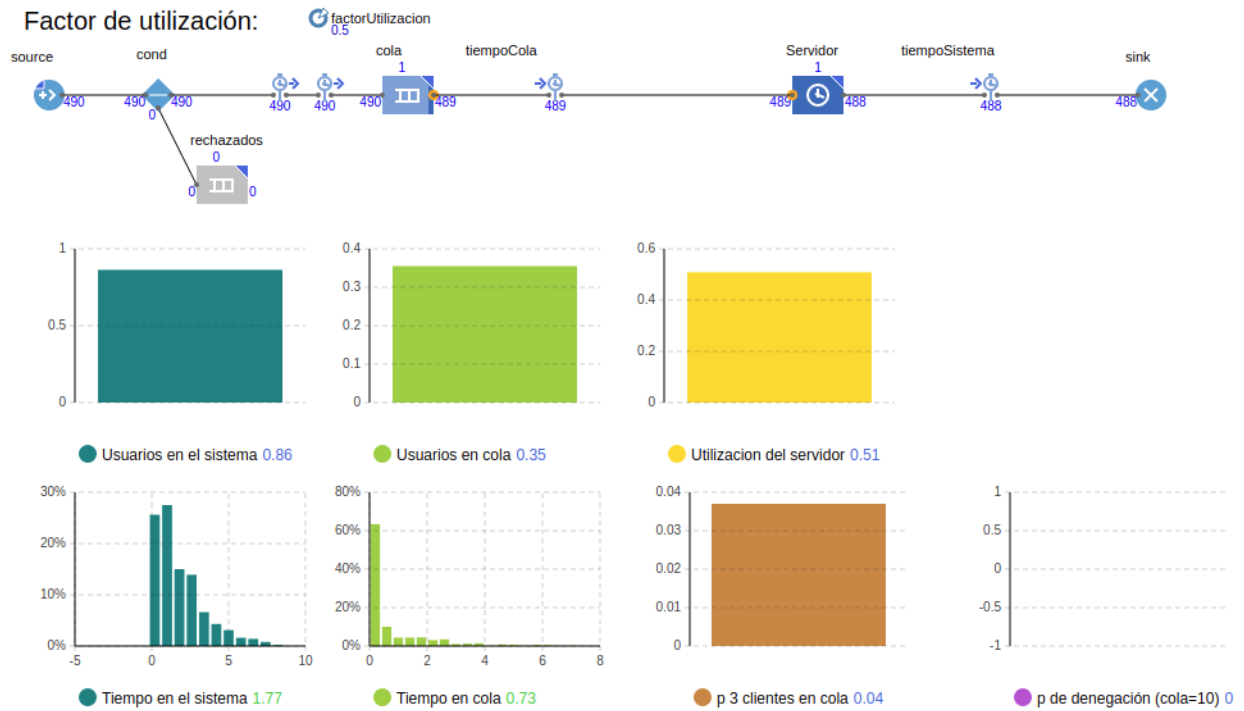


Figure 4: Comparación de resultados de AnyLogic para una tasa de arribo del 50%.

En la Figura 4, se muestran los resultados obtenidos con AnyLogic luego de varias corridas eligiendo un escenario promedio.

Los resultados de AnyLogic muestran estar un poco alejados de los valores teóricos para todas las métricas analizadas. Las diferencias observadas entre los resultados simulados en Python y los obtenidos con AnyLogic son considerables aunque ambos enfoques siguen siendo válidos para modelar el sistema M/M/1 bajo una tasa de arribo del 50%.

Conclusión: Para una tasa de arribo del 50%, se observa una alta concordancia entre los resultados teóricos y los simulados en Python y una pequeña diferencia con AnyLogic. Pero en general, todos los métodos confirman la validez del modelo M/M/1 bajo estas condiciones.

3.1.3 Tasa de arribo: 75%

Vamos a analizar el comportamiento del sistema de colas M/M/1 con una tasa de arribo correspondiente al : 75% de la tasa de servicio. Las métricas analizadas van a ser las mencionadas en la introducción, y se van a comparar a partir de 3 fuentes : la simulación en Python, los valores teóricos calculados y los resultados obtenidos con AnyLogic.

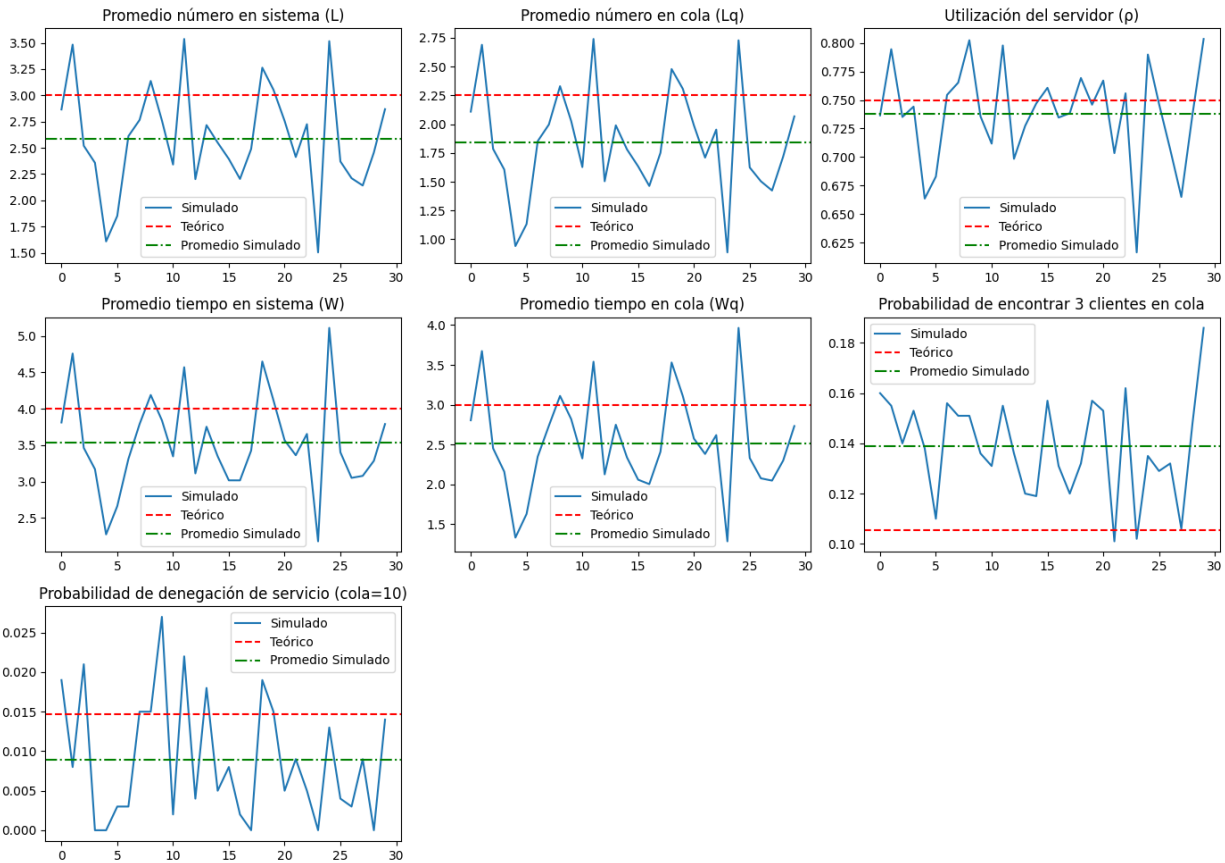


Figure 5: Comparación de resultados simulados en Python y valores teóricos para una tasa de arribo del 75%.

En la Figura 5, se presentan los resultados obtenidos a partir de la simulación en Python comparados con los valores teóricos calculados para un sistema M/M/1 con una tasa de arribo del 75% de la tasa de servicio.

Para el número promedio de clientes en el sistema (L), los resultados simulados en Python muestran una variabilidad significativa alrededor del valor teórico de 3.0. Aunque los valores simulados fluctúan, el promedio de estas simulaciones (línea verde punteada) se aproxima bastante al valor teórico, indicando que, en promedio, la simulación refleja correctamente el comportamiento esperado del sistema.

El número promedio de clientes en cola (L_q) sigue un patrón similar, con variaciones en los resultados simulados que se distribuyen alrededor del valor teórico de 2.25. Esta variabilidad es un reflejo de la naturaleza estocástica del proceso de arribo y servicio en el sistema de colas.

La utilización del servidor (ρ) es una métrica crucial, y en este caso, los valores simulados se alinean bastante bien con el valor teórico de 0.75, aunque con algunas fluctuaciones. Esto sugiere que la simulación es efectiva para capturar la dinámica de uso del servidor bajo una carga significativa.

El tiempo promedio en el sistema (W) y el tiempo promedio en cola (W_q) presentan oscilaciones en la simulación en torno a los valores teóricos de 4.0 y 3.0, respectivamente. Estas fluctuaciones pueden atribuirse a la variabilidad natural en la simulación, pero el promedio de los resultados simula adecuadamente los tiempos esperados.

Finalmente, las probabilidades de encontrar 3 clientes en cola y la probabilidad de denegación de servicio muestran una consistencia general entre la simulación y los valores teóricos. Las pequeñas diferencias observadas se deben a la variabilidad inherente en los experimentos de simulación estocástica.

Comparación con AnyLogic:



Figure 6: Comparación de resultados de AnyLogic para una tasa de arribo del 75%.

En la Figura 6, se muestran los resultados obtenidos con AnyLogic luego de varias corridas eligiendo un escenario promedio.

Los resultados obtenidos con AnyLogic muestran una alta precisión y estabilidad, con una coincidencia casi exacta con los valores teóricos para todas las métricas analizadas. A diferencia de los resultados simulados en Python, los resultados de AnyLogic presentan menos fluctuaciones y mayor consistencia, lo que sugiere que AnyLogic maneja de manera más efectiva la aleatoriedad y la variabilidad en el sistema.

Por ejemplo, en el caso del número promedio de clientes en el sistema (L) y el número promedio de clientes en cola (L_q), AnyLogic logra replicar los valores teóricos con gran precisión, reduciendo considerablemente la variabilidad observada en Python.

Además, la utilización del servidor (ρ) y las métricas de tiempo (W y W_q) en AnyLogic son prácticamente idénticas a los valores teóricos, confirmando su superioridad en la simulación de sistemas de colas.

Conclusión:

Para una tasa de arribo del 75%, se observa una excelente concordancia entre los resultados teóricos y los obtenidos con AnyLogic, lo que contrasta ligeramente con la mayor variabilidad observada en las simulaciones realizadas en Python. Aunque ambas herramientas confirman la validez del modelo M/M/1, AnyLogic ofrece resultados más precisos y consistentes.

3.1.4 Tasa de arribo: 100%

El análisis de un sistema de colas M/M/1 con una tasa de arribo equivalente al 100% de la tasa de servicio es particularmente interesante debido a la condición crítica en la que se encuentra el sistema. En este caso, el sistema opera en su límite máximo de capacidad, lo que implica un comportamiento significativamente diferente al de tasas de arribo menores. A continuación, se comparan los resultados de las simulaciones realizadas en Python con los valores teóricos calculados.

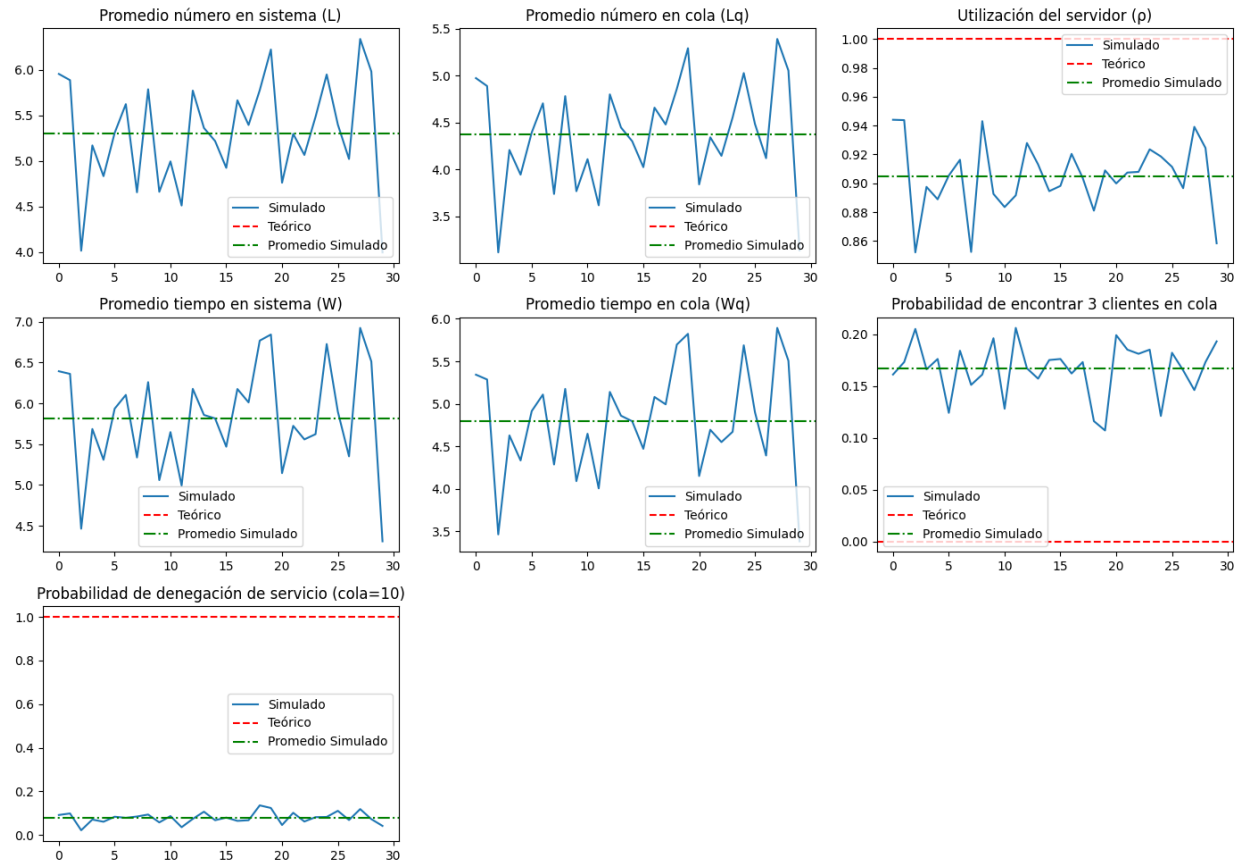


Figure 7: Comparación de resultados simulados en Python y valores teóricos para una tasa de arribo del 100%.

En la Figura 7, se observa que para el número promedio de clientes en el sistema (L), los resultados simulados en Python presentan una variabilidad significativa alrededor del promedio simulado, representado por la línea verde punteada. Cabe destacar que el valor teórico para L en este escenario tiende al infinito debido a que, con una tasa de arribo igual a la tasa de servicio, el sistema se encuentra en un estado donde cualquier pequeña fluctuación puede causar una acumulación indefinida de clientes en el sistema. Esto explica la ausencia de la línea roja teórica en la gráfica.

El número promedio de clientes en cola (L_q) sigue un patrón similar, con resultados simulados que oscilan alrededor del promedio simulado. Al igual que en el caso de L , el valor teórico de L_q también tiende al infinito bajo estas condiciones de operación. La alta variabilidad observada en esta métrica refleja la sensibilidad extrema del sistema cuando se encuentra al borde de la saturación, donde cualquier perturbación en los tiempos de arribo o servicio puede generar un incremento significativo en la longitud de la cola.

En lo que respecta a la utilización del servidor (ρ), los resultados simulados muestran un alineamiento cercano con el valor teórico de 1.0. Aquí, la línea roja teórica está presente pero coincide con el valor máximo posible de la métrica, indicando que el servidor está operando al 100% de su capacidad. Las pequeñas fluctuaciones observadas en los resultados simulados se deben a la variabilidad inherente en la simulación, pero el promedio simulado (línea verde punteada) confirma que la simulación captura correctamente la utilización completa del servidor.

El tiempo promedio en el sistema (W) y el tiempo promedio en cola (W_q) presentan un comportamiento análogo, con valores simulados que fluctúan en torno al promedio simulado. En este contexto, es importante destacar que las fórmulas teóricas para estos tiempos sugieren que, bajo una tasa de arribo igual a la tasa de servicio, los tiempos teóricos tienden al infinito. Esto se debe a que, en un sistema saturado, el tiempo de espera para los clientes puede crecer indefinidamente. La ausencia de la línea roja teórica en estas gráficas refleja precisamente este hecho.

La probabilidad de encontrar 3 clientes en cola muestra una consistencia general entre la simulación y el promedio simulado, aunque con algunas oscilaciones menores, lejos del teórico 0. Por otro lado, la probabilidad de denegación de servicio, calculada para un límite de 10 clientes en cola, es sustancialmente baja en la simulación en comparación con

la expectativa teórica. Esto sugiere que, incluso bajo una carga total, el sistema no alcanza frecuentemente el umbral de rechazo de clientes, lo cual podría deberse a la forma en que se maneja la variabilidad en la simulación estocástica.

Comparación con AnyLogic



Figure 8: Comparación de resultados de AnyLogic para una tasa de arribo del 100%.

Las simulaciones realizadas en AnyLogic han demostrado una notable coherencia con los resultados obtenidos a través del programa en Python, variando muy poco entre los parametros. Sin embargo, se observa que tanto los resultados de AnyLogic como los de Python se desvían significativamente de los valores teóricos esperados, lo que sugiere la presencia de factores o simplificaciones que podrían estar afectando la precisión en ambos casos.

Conclusión:

Las discrepancias observadas entre los resultados de las simulaciones en AnyLogic y Python respecto a los valores teóricos pueden explicarse por el hecho de que ambos modelos tienden a ajustarse a un índice de utilización cercano a 0.92, en lugar de alcanzar el valor teórico de 1. Esto indica que, aunque el sistema se acerca a su capacidad máxima, sigue atendiendo ligeramente más rápido que el ritmo de arribo, lo que genera tiempos de espera menores a los esperados en un escenario teórico de plena utilización.

3.1.5 Tasa de arribo: 125%

El análisis de un sistema de colas M/M/1 con una tasa de arribo equivalente al 125% de la tasa de servicio presenta un caso extremo donde el sistema se encuentra en un estado de sobrecarga. Este escenario lleva a resultados teóricos que son divergentes, lo que se observa en las gráficas comparativas entre los valores simulados y los teóricos.

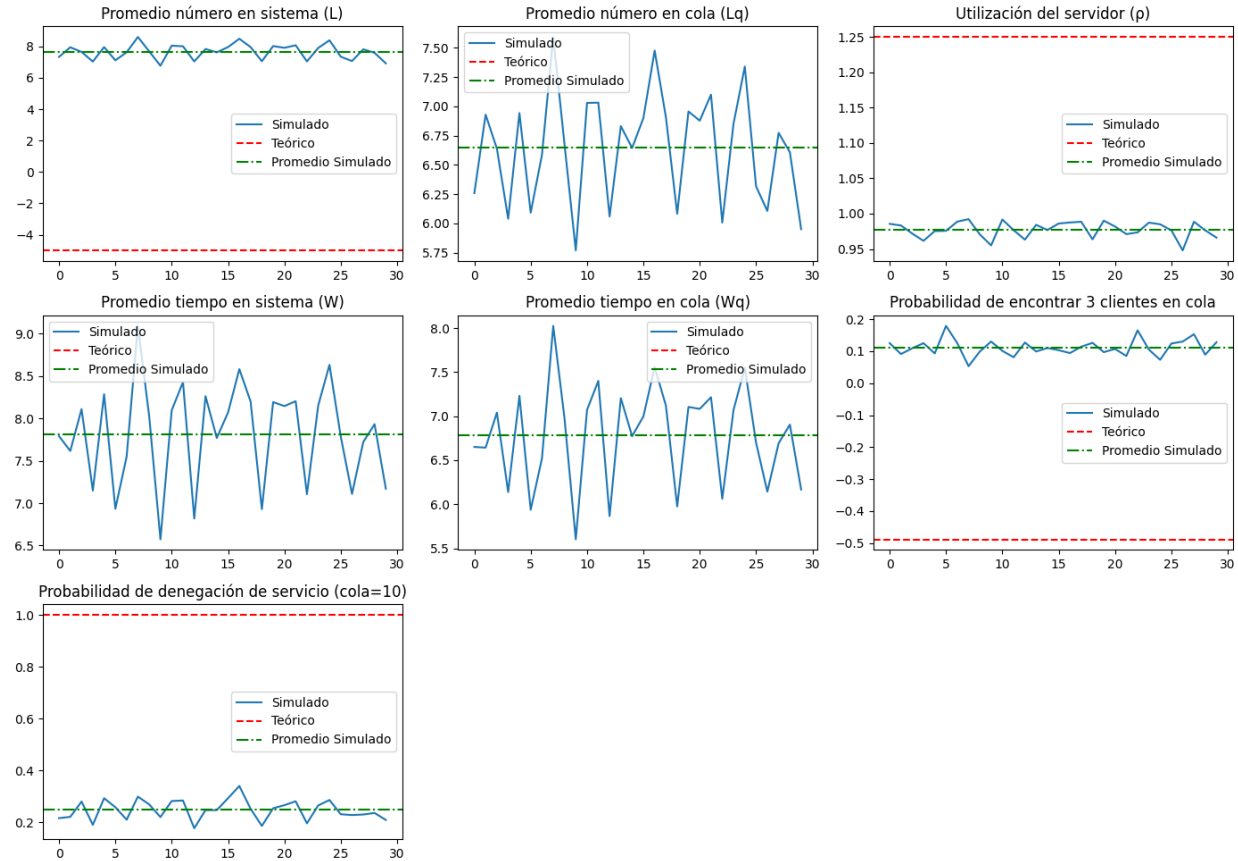


Figure 9: Comparación de resultados simulados en Python y valores teóricos para una tasa de arribo del 125%.

En la Figura 11, el número promedio de clientes en el sistema (L) muestra una variabilidad considerable en los resultados simulados, con valores que oscilan alrededor del promedio simulado (línea verde punteada). Sin embargo, la línea roja teórica no se encuentra en la gráfica porque el valor teórico tiende al infinito. Esto es debido a que en un sistema donde la tasa de arribo excede la tasa de servicio, la longitud de la cola y el número de clientes en el sistema aumentan indefinidamente con el tiempo, lo que genera un crecimiento no acotado de L .

Para el número promedio de clientes en cola (L_q), se observa un comportamiento similar. Los resultados simulados fluctúan alrededor de un promedio finito, pero el valor teórico de L_q también tiende al infinito. Esto refleja el hecho de que, teóricamente, cuando el sistema está saturado más allá de su capacidad (con una tasa de arribo mayor al 100% de la tasa de servicio), las colas se acumulan indefinidamente, llevando a una divergencia en los valores teóricos.

La utilización del servidor (ρ) en los resultados simulados se aproxima a 1.0, aunque presenta ligeras fluctuaciones que son un intento del sistema de mantener un equilibrio bajo condiciones extremas. La línea roja teórica en esta gráfica está ubicada en 1.25, lo que refleja la condición en la que el sistema está trabajando más allá de su capacidad teórica (es decir, el servidor debería estar operando al 125% de su capacidad). Sin embargo, en la práctica, el servidor no puede exceder el 100% de utilización, lo que explica por qué el promedio simulado no llega al valor teórico de 1.25. El programa de Python "protege" al sistema de esta sobreutilización, limitando la utilización al 100%, lo cual es físicamente razonable.

Los tiempos promedio en el sistema (W) y en cola (W_q) también exhiben fluctuaciones en torno al promedio simulado, pero sin alcanzar el infinito como sugieren los valores teóricos. Estos tiempos, bajo un escenario de sobresaturación, deberían crecer sin límite, dado que los clientes experimentan tiempos de espera cada vez mayores. La simulación, sin embargo, muestra que el sistema alcanza un punto donde el tiempo de espera se estabiliza a un valor alto pero finito, lo que nuevamente indica una protección inherente en la simulación que previene la explosión indefinida de estos tiempos.

La probabilidad de encontrar 3 clientes en cola se vuelve teóricamente negativa por una situación donde reemplazando en la fórmula nos queda una resta donde el sustraendo es mayor que el minuendo. Por otro lado, la probabilidad de denegación de servicio para un límite de 10 clientes en cola muestra un valor notablemente más bajo en la simulación

en comparación con el valor teórico. Esto sugiere que, aunque el sistema está sobresaturado, la simulación maneja la cola de manera que raramente alcanza el límite máximo, probablemente debido a la misma lógica de protección que limita la utilización del servidor.

Comparación con AnyLogic



Figure 10: Comparación de resultados de AnyLogic para una tasa de arribo del 125%.

Las simulaciones realizadas en AnyLogic han demostrado una notable coherencia con los resultados obtenidos a través del programa en Python, variando muy poco entre los parametros (en algunos mas que en otros). Sin embargo, se observa que tanto los resultados de AnyLogic como los de Python se desvían significativamente de los valores teóricos esperados. Aunque se llega a llenar y a rechazar usuarios (0.22 de probabilidad de denegación)

Conclusión:

Las discrepancias observadas entre los resultados de las simulaciones en AnyLogic y Python respecto a los valores teóricos pueden explicarse por el hecho de que ambos modelos tienden a ajustarse a un índice de utilización cercano a 0.99, en lugar de alcanzar el valor teórico de 1.25. Esto indica que, al igual que en el ejemplo anterior, aunque el sistema se acerca a su capacidad máxima, sigue atendiendo ligeramente más rápido que el ritmo de arribo, lo que genera tiempos de espera menores a los esperados en un escenario teórico de plena utilización por lo que nos da valores reales de usuarios en el sistema y usuarios en cola.

3.2 Modelo de inventario

En esta sección se analizará el comportamiento del inventario variando el tiempo de simulación (en meses), el tiempo promedio entre demandas consecutivas, el costo de reabastecimiento, el costo de realizar un pedido, la cantidad mínima de productos para reabastecerse, la cantidad de productos que quieres tener cuando te reabasteces y la cantidad de productos con los que inicias. Se analizará el costo de orden, el costo de mantenimiento, el costo de faltante y el costo total, que es la suma de los tres anteriores.

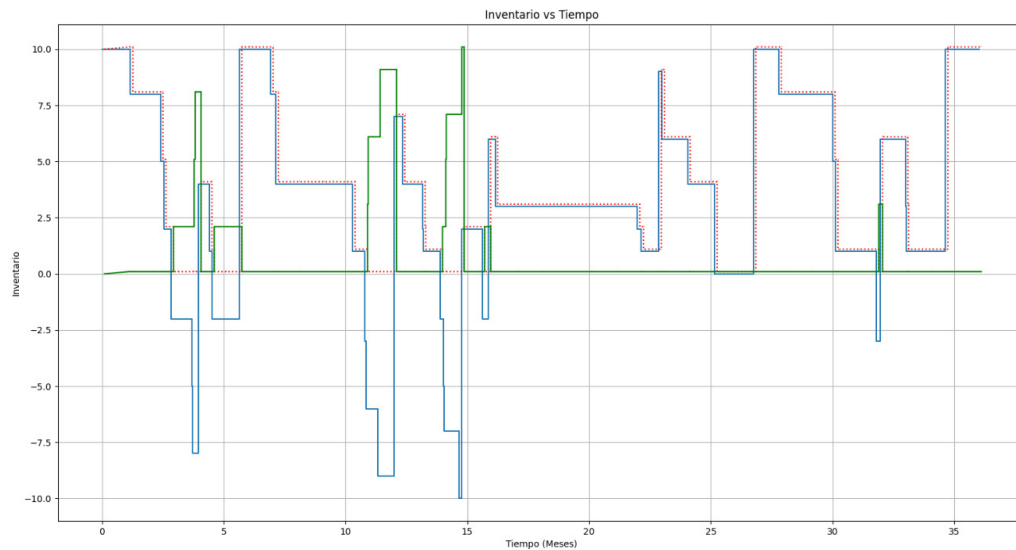


Figure 11: Resultados simulados en python . $(t) = 36$, $(s) = 3$, $(S) = 10$ y $\lambda = 0.8$

Podemos observar en la Figura 11 que a medida que transcurre el tiempo, el inventario se va agotando debido a la demanda continua. Cuando el nivel de productos en inventario desciende hasta el umbral mínimo de 3 unidades o menos, se produce un evento de reabastecimiento que eleva el inventario nuevamente hasta el máximo de 10 unidades. Este ciclo de agotamiento y reabastecimiento se repite a lo largo del tiempo, formando un patrón de oscilación característico de los sistemas gestionados bajo políticas (s, S) .

Sin embargo, el gráfico también revela momentos en los que el inventario cae por debajo de cero, lo que sugiere que en algunos períodos la demanda superó la capacidad de respuesta del sistema, generando situaciones de faltantes. Estos períodos de inventario negativo indican una insuficiencia en el suministro para satisfacer la demanda, lo que podría derivar en costos asociados a la falta de productos, como la pérdida de ventas o la insatisfacción del cliente.

Tras cada evento de reabastecimiento, el inventario vuelve a alcanzar el nivel máximo de 10 unidades, lo que resalta la naturaleza cíclica del proceso de gestión de inventarios en este modelo. La capacidad del sistema para recuperar su inventario hasta el nivel deseado tras cada ciclo de agotamiento subraya la eficacia de la política (s, S) para mantener un equilibrio entre el nivel de servicio y la minimización de los costos de almacenamiento.

3.2.1 Parámetros

Durante todo el desarrollo utilizaremos los siguientes parámetros base y variaremos los siguientes parámetros y observaremos los resultados de simular dicho inventario.

Tiempo de simulación (en meses) (t) : 36

Demanda (λ) : 0.8

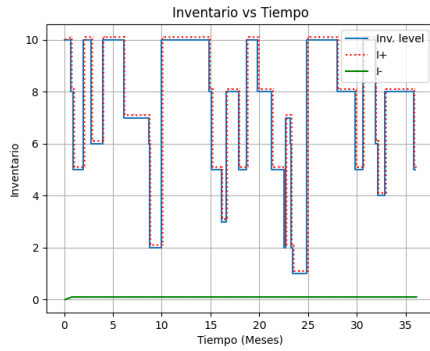
Cantidad mínima de productos para reabastecerse (s) : 3

Capacidad del inventario (S) : 10

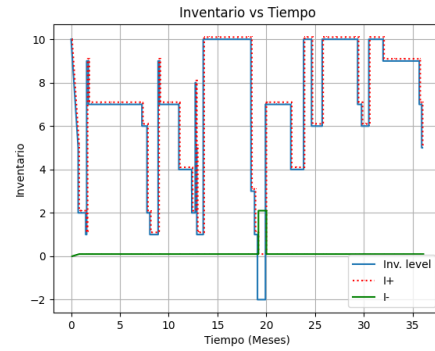
Puesto que la variable s es sobre la que tenemos mayor control en una situación real, realizaremos simulaciones variando solo ese parámetro, con el objetivo de encontrar valores óptimos para una variedad de situaciones.

3.2.2 Modelo con $s = 7$

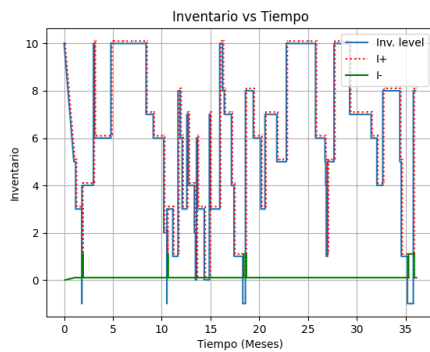
La primera simulación que realizamos es con $s=7$. Es decir que no se espera a tener una cantidad relativa baja de inventario para realizar un pedido. Vamos a observar una muestra reducida de 4 gráficas de niveles de inventario, ya que no tendría sentido calcular los valores de inventario promedio. El análisis de los estadísticos se realiza más adelante sobre una muestra mayor.



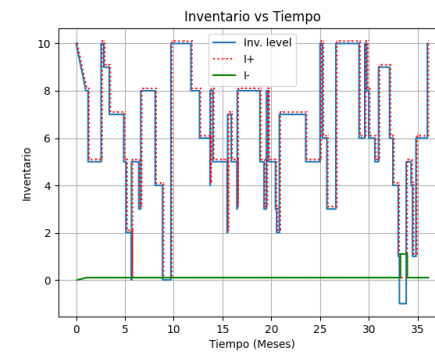
(a) 1a



(b) 1b



(c) 1b



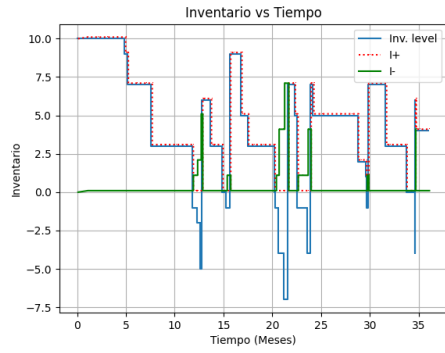
(d) 1b

 Figure 12: Niveles de inventario con $s=7$

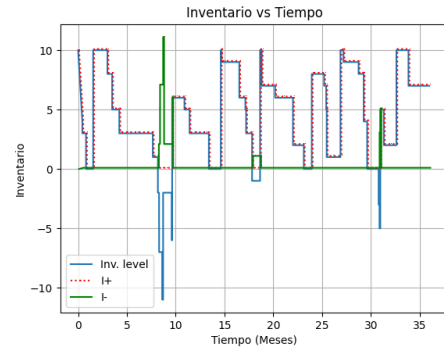
Se observa que con un nivel de pedido de $s=7$ el inventario casi nunca llega a estar en valores negativos. Esto podría ser útil si no se puede permitir la venta de artículos sin existencias o si el costo de escasez (p) es muy alto en relación al costo de mantenimiento. También observamos que la cantidad de artículos reales en el inventario ($I+$) es considerablemente alta en todo momento, lo cual podría provocar gastos innecesarios si el costo de mantenimiento (h) es demasiado alto. Analizamos los costos variando los parámetros h y p en la sección ??.

3.2.3 Modelo con $s = 3$

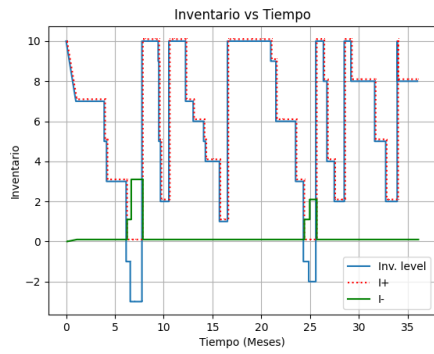
Configurando un parámetro s menor se espera más tiempo en promedio antes de realizar el siguiente pedido. Esto lleva a más situaciones con nivel de inventario negativo. Dependiendo de los valores de p y h podría provocar un costo total mayor. En general el costo de escasez tendrá a ser más grande que el de mantenimiento, por lo que no pareciera ser buena idea reducir el s a 3.



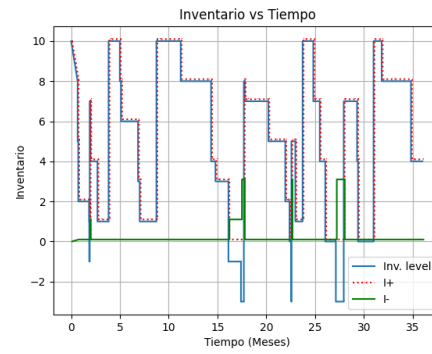
(a) 1a



(b) 1b



(c) 1b



(d) 1b

Figure 13: Niveles de inventario con $s = 3$

3.3 Análisis de costos

3.3.1 Análisis con $s=0$

Primero vamos a estudiar el sistema cuando ($s=0$). Donde el pedido se realiza una vez que el inventario real llega a 0.

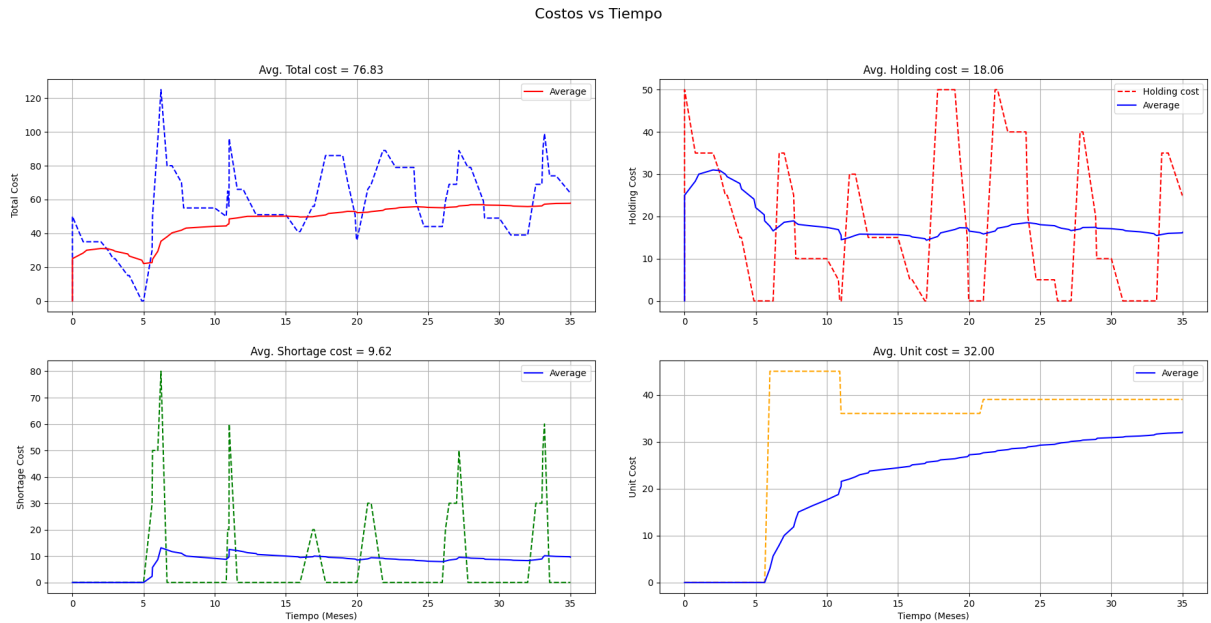


Figure 14: Simulación con parámetros de costos (k) = 7, (p) = 10, (h) = 5 y i = 3

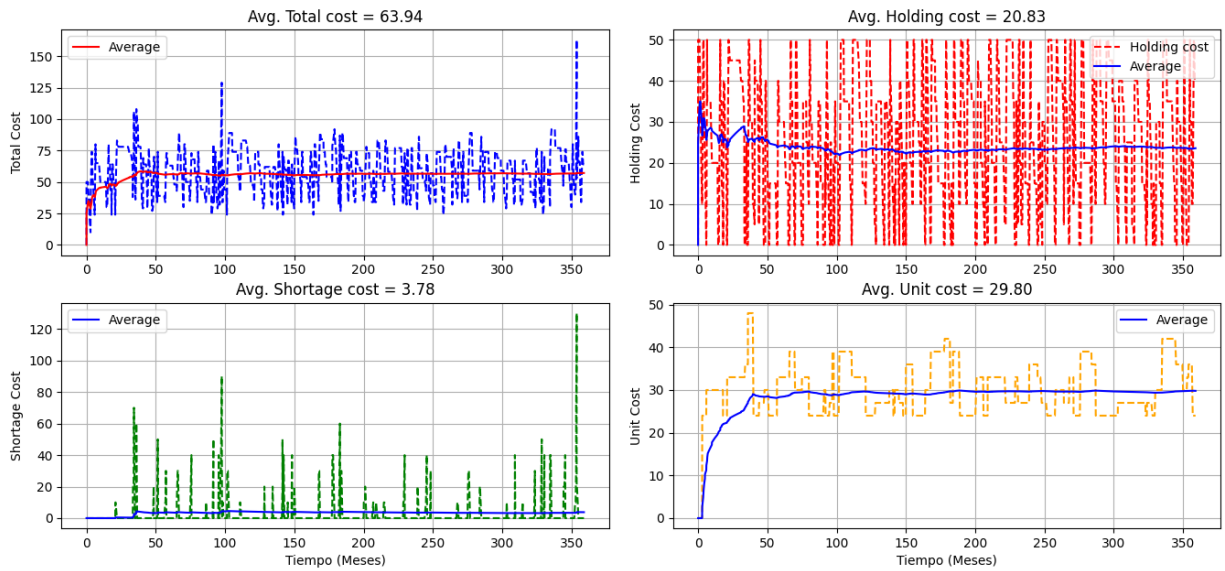
En la Figura 14 se observa que la mayor contribución al costo total está dada por el costo unitario del producto (con un promedio de 32) y una contribución del 41.65%, apesar de que el costo unitario sea el más bajo de los 4 parámetros. Como $s = 0$, la cantidad pedida en cada iteración seará muy grande ($S - s = S$) y por ende el costo representará una gran parte del total.

Se observa también que el costo de escasez de producto es el que menos afecta al costo total, con aproximadamente un 12.52%, apesar de que el parámetro p tenga un valor considerablemente grande y el I — esté maximizado (porque $s = 0$). Podemos hipotizar entonces, que siempre es recomendable tener una gran parte del inventario negativo, ya que aquí tenemos la situación menos favorable y aún así resulta tener el menor aporte al costo total.

3.3.2 Analisis con $s=3$

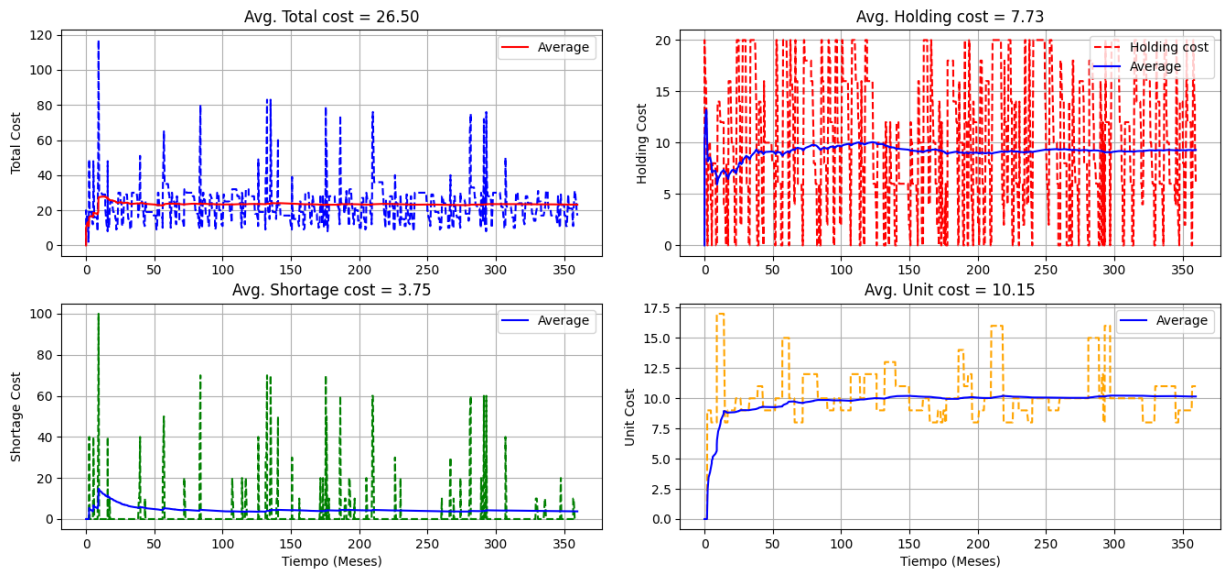
Realizaremos una simulación aumentando el tiempo de simulación a 30 años (36 meses) Con el fin de obtener promedios de costos más fieles a la realidad del sistema simulado.

Costos vs Tiempo


 Figure 15: Simulación con parámetros de costos $(k) = 7$, $(p) = 10$, $(h) = 5$ y $i = 3$

En la figura 15 se observa que el costo total se redujo ligeramente a comparación de la simulación 3.3.1. El costo unitario sigue siendo el costo que más aporta al costo total. Debido a que tenemos una s pequeña, tendremos unidades en inventario la mayoría del tiempo, por lo que reducir el costo de holding reduce considerablemente el costo total. Reduciendo h y i obtenemos:

Costos vs Tiempo


 Figure 16: Simulación con parámetros de costos $(k) = 7$, $(p) = 10$, $(h) = 2$ y $i = 1$

En la figura 16 observamos que el costo se vió considerablemente reducido (de 64.94 a 26.50, casi un 50%) al reducir los costos de holding y de unidad.

4 Conclusiones

La simulación del modelo MM1 reveló que el promedio de clientes en el sistema y en la cola aumenta significativamente cuando la tasa de arribo se aproxima a la tasa de servicio. Este comportamiento se alinea con las expectativas teóricas del modelo, donde una mayor tasa de llegada en comparación con la tasa de servicio conduce a una mayor acumulación de clientes. Además, tanto el tiempo promedio en el sistema como en cola también aumentó con la tasa de arribo, lo que indica una mayor congestión en el sistema. La utilización del servidor también creció con la tasa de arribo, acercándose a 1 cuando la tasa de arribo iguala o supera la tasa de servicio, sugiriendo que el servidor está casi siempre ocupado. Las probabilidades de encontrar diferentes números de clientes en la cola se ajustaron a las fórmulas teóricas del modelo MM1, y la probabilidad de denegación de servicio aumentó cuando el tamaño de la cola finita se redujo, reflejando el impacto de limitar la capacidad de la cola en el sistema.

En cuanto a las variaciones en las tasas de arribo, los resultados confirmaron que estas tienen un impacto notable en las medidas de rendimiento. A mayores tasas de arribo en relación con la tasa de servicio, se observó un aumento en la congestión del sistema y en los tiempos de espera. Los resultados de la simulación fueron coherentes con los valores teóricos esperados del modelo MM1 y también mostraron alta precisión en las métricas de rendimiento, confirmando la robustez del modelo en distintos contextos de simulación.

Para el modelo de inventario, la simulación mostró que el costo total del inventario se compone principalmente del costo de mantenimiento y del costo de faltante. Ajustar los parámetros del modelo es crucial para equilibrar estos costos y minimizar el costo total. Los resultados indicaron cómo varían los costos totales con diferentes parámetros del modelo, demostrando que las decisiones sobre los niveles de inventario y las políticas de orden tienen un impacto significativo en el costo total. Los resultados de la simulación coincidieron en general con las expectativas teóricas y proporcionaron una visión detallada de los costos asociados al modelo. La simulación en distintos entornos confirmó la validez del modelo y mostró consistencia con los valores teóricos y simulados. Dadas las simulaciones realizadas para diferentes costos podemos concluir que elegir un valor de s y S es muy importante para obtener un mínimo óptimo de costos totales. Para sistemas en los que el costo de escasez es muy grande en relación a los costos de mantenimiento, es conveniente tener valores de s y S que estén cercanos entre sí. De este modo, se reducen los costos de tener inventario negativo y los costos de escasez relacionados con ello. Este tipo de sistema encaja perfectamente con productos de venta on-line donde tener productos en stock tiene un costo muy bajo o incluso nulo (si se trata de un producto digital). Para sistemas donde el costo de escasez no es demasiado más grande que los costos de mantenimiento y unitario es conveniente tener valores de s y S más cercanos entre sí. Porque de este modo valoramos más tener inventario negativo, que es más barato, reduciendo así los costos.

En resumen, la simulación de ambos modelos proporcionó una comprensión profunda de las dinámicas de los sistemas estudiados. La comparación entre resultados teóricos y simulaciones en diferentes entornos demostró la eficacia y precisión de los modelos. Se recomienda ajustar los parámetros según las necesidades específicas del sistema para lograr un equilibrio óptimo entre costos y eficiencia en aplicaciones prácticas.