

# **Informe del Proyecto 1 de Simulación Inventario**

Integrantes:

Juan Carlos Espinosa Delgado C-411

Alex Sierra Alcalá C-411

# 1 Introducción

## 1.1 Breve descripción del proyecto

El proyecto tiene como objetivo desarrollar una simulación de eventos discretos para analizar y comprender mejor el comportamiento de un sistema de inventario en un entorno comercial. Se explorará cómo un tendero gestiona su inventario para satisfacer las demandas de los clientes, utilizando una política de pedido específica y teniendo en cuenta diversos costos asociados.

## 1.2 Objetivos y metas

- Comprender el funcionamiento de un sistema de inventario bajo la política de pedido  $(s, S)$ .
- Analizar el impacto de diferentes parámetros, como los puntos de reorden y los costos de pedido y mantenimiento, en el rendimiento del inventario.
- Desarrollar una simulación de eventos discretos que modele de manera efectiva el comportamiento del inventario y permita realizar experimentos para evaluar su desempeño.

## 1.3 Sistema específico a simular y variables de interés

Las variables de interés incluyen:

- Punto de reorden  $(s)$ .
- Cantidad de pedido  $(S)$ .
- Demanda de clientes.
- Costo de pedido y unidades del producto  $(c(y))$ .
- Costo de mantenimiento de inventario por unidad de tiempo  $(h)$ .
- Tiempo de entrega de pedidos  $(L)$ .
- Pérdida por insatisfacción de la demanda del cliente.
- Distribución de la demanda.
- Distribución del tiempo entre llegadas de clientes.

# 2 Detalles de implementación

Para la implementación de la simulación se utilizó el lenguaje de programación Python, así como las librerías numpy, scipy, pandas y matplotlib para el manejo de datos y la visualización de resultados.

- Clase InventorySimulation:
  - Modelado de la simulación: El primer paso fue definir cómo modelar el sistema de inventario en términos de objetos y comportamientos. Se creó una clase InventorySimulation que actúa como el controlador principal de la simulación. Esta clase gestiona el tiempo de simulación, la cola de eventos y las interacciones con el inventario.

- Abstracción de eventos: Se utilizó una estructura de cola de eventos para modelar los eventos discretos en el sistema, como la llegada de clientes, los pagos diarios y la entrega de pedidos. Cada evento se representa como un objeto con un tiempo asociado y una acción a realizar en ese momento.
  - Dinámica del inventario: Se diseñó una clase Inventory para representar el inventario y su estado en cada momento. Esta clase gestiona la cantidad de artículos disponibles, los pedidos pendientes y los costos asociados con el mantenimiento y la reposición del inventario.
  - Simulación del flujo de clientes: Se modeló la llegada de clientes y la demanda de productos utilizando distribuciones de probabilidad como la distribución de Poisson y la distribución uniforme discreta. Esto permitió simular el comportamiento aleatorio de los clientes y su efecto en el inventario.
  - Análisis de resultados: Se implementaron métodos para recopilar y analizar los resultados de la simulación, como el número de clientes atendidos, la cantidad de productos vendidos y los costos asociados con el mantenimiento del inventario. Esto permitió evaluar el rendimiento del sistema y comparar diferentes estrategias de gestión de inventario.
  - Visualización de resultados.
- Clase Inventory:
    - Representación del inventario: Se diseñó una clase para representar el inventario y su estado en un momento dado. Esto incluyó la cantidad de artículos disponibles, los pedidos pendientes y los costos asociados con el mantenimiento y la reposición del inventario.
    - Gestión de lógica de inventario: Se implementaron métodos para gestionar la llegada de pedidos, la entrega de productos, el mantenimiento del inventario y la evaluación de la demanda insatisfecha. Esto permitió simular el comportamiento del inventario y evaluar diferentes estrategias de gestión.
  - Clase InventoryState:
    - Representación del estado del inventario: Se diseñó una clase para representar el estado del inventario en un momento dado. Esto permitió mantener un registro de la cantidad de artículos en inventario y si había pedidos pendientes en un momento dado, lo que facilitó el seguimiento y la gestión del inventario durante la simulación.

## 3 Resultados y experimentos

Todos los códigos empleados para llegar a estos resultados están

### 3.1 Comparando simulaciones alternativas

En este apartado compararemos el rendimiento de dos sistemas para determinar cuál funciona mejor. En general, ejecutar una sola réplica de cada sistema no es una buena práctica, ya que la respuesta será aleatoria y corremos el riesgo de tomar una mala decisión. Tenemos entonces varios enfoques posibles para este problema. Primero vamos a ejecutar varias réplicas independientes de cada sistema y, a continuación, vamos a crear un intervalo de confianza para la diferencia por pares en los costos totales.

Partiremos entonces de las siguientes suposiciones iniciales para algunos parámetros del sistema:

- Cantidad inicial de productos en inventario: 40
- Función de petición de productos al distribuidor:  $c(y) = 3y + 30$
- Tiempo de demora de llegada después de una petición: 0.75 unidades de tiempo

- Costo de almacenamiento de un producto por unidad de tiempo: 1
- Distribución de la demanda de los clientes: Uniforme entre 1 y 5
- Distribución del tiempo entre llegadas de clientes: Exponencial con  $\lambda = 0.05$

Vamos a comparar dos configuraciones:

- Sistema 1:  $(s, S) = (20, 50)$
- Sistema 2:  $(s, S) = (20, 80)$

Para esto realizamos 30 simulaciones independientes para tener en cuenta la aleatoriedad en el modelo. El costo promedio de nuestra segunda configuración es aproximadamente un 4 por ciento más bajo que la primera configuración de nuestra muestra. Si bien esto es significativo en la práctica, ya que podría resultar en ahorros sustanciales para una empresa, realizamos posteriormente un análisis más sólido para asegurarnos de que no estamos observando una diferencia puramente debido a la aleatoriedad. Primero visualicemos los datos que hemos generado para ambas configuraciones.

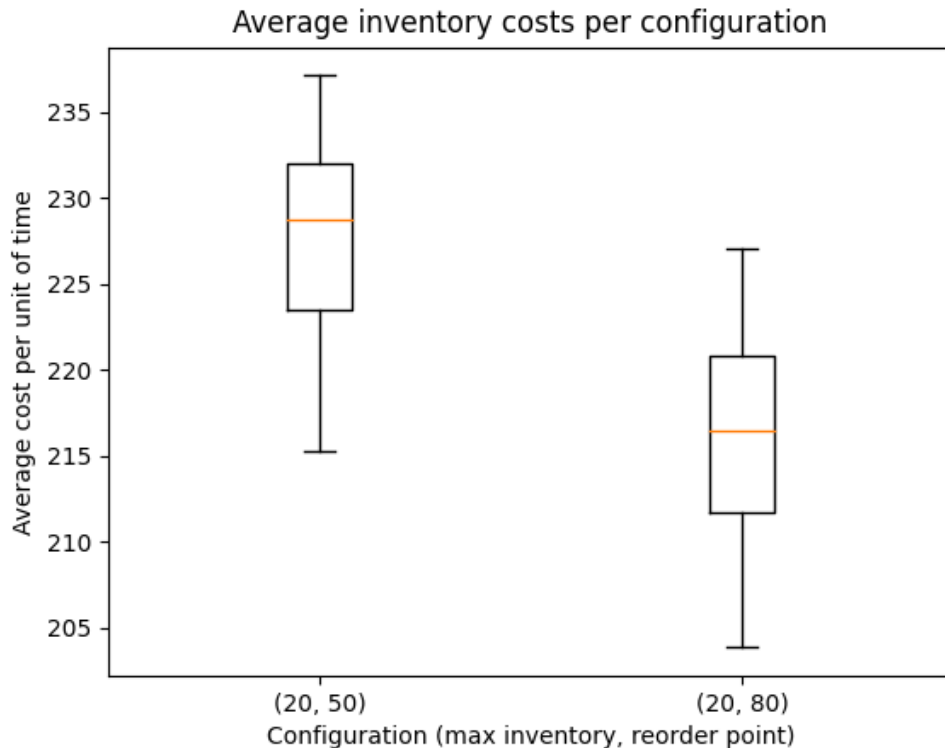


Figure 1: Promedio de costos totales para las dos configuraciones

En general, los costos de la segunda configuración parecen ser más bajos que los de la primera configuración. Sin embargo, todavía existe una superposición considerable entre los resultados de ambos sistemas.

Formulemos dos hipótesis:

- $H_0$  Hipótesis nula: (coste del sistema 1 - costo del sistema 2) = 0

–  $H_1$  Hipótesis alternativa: (coste del sistema 1 - costo del sistema 2)  $\neq 0$

Cuando analizamos el rendimiento de un solo sistema, construimos un intervalo de confianza para el costo promedio por unidad de tiempo. Vamos a aplicar un enfoque similar para construir un intervalo de confianza del 90 por ciento de la diferencia de costos entre ambos sistemas para ver si 0 está incluido en el IC. Si es así, significa que no podemos rechazar la hipótesis nula de que ambas medias son iguales. Vamos a emparejar observaciones de ambos sistemas y restar los promedios de costos de cada par para poder construir un intervalo de confianza para la diferencia. Para nuestros datos, el intervalo de confianza de la diferencia de costos quedó entre [7.7, 12.3].

Como El IC del 90 por ciento para la diferencia de costos no incluye 0, podemos afirmar que no se cumple  $H_0$  y concluir que el costo promedio es significativamente menor para la configuración 2.

### 3.2 Minimizando los costos del inventario

Ahora exploraremos el resultado del modelo de simulación para determinar la mejor combinación de punto de reorden e inventario máximo que minimice el costo total por unidad de tiempo del sistema de inventario.

Una forma de encontrar una combinación de entradas que optimice una respuesta es ejecutar varias repeticiones del modelo de simulación en una amplia gama de configuraciones posibles y analizar cada una de las salidas individuales. En particular, vamos a ejecutar todas las configuraciones con puntos de reorden que van de 0 a 120 e inventario máximo de 10 a 150 con incrementos de 10. Cada configuración se ejecutará 10 veces.

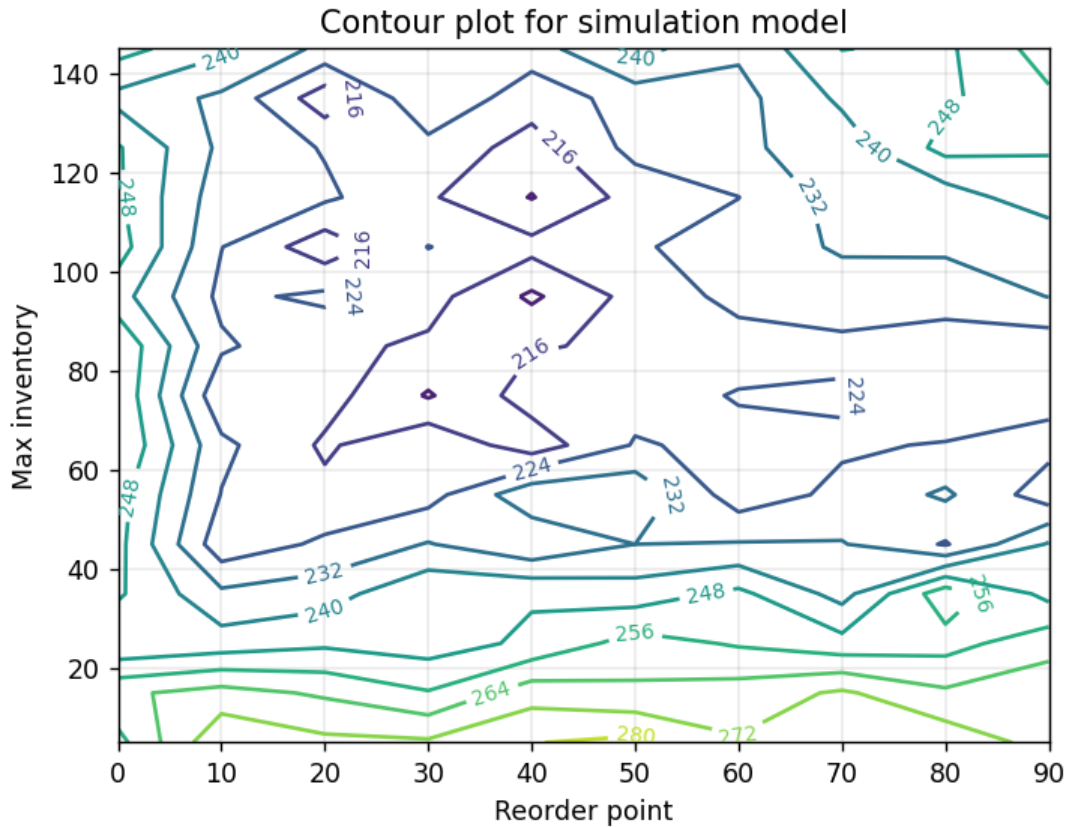


Figure 2: Promedio de costos totales por unidad de tiempo

El gráfico de contorno nos muestra el costo total promedio por unidad de tiempo esperado para cada combinación de inventario máximo y punto de reorden.

De los datos podemos analizar que el costo mínimo del sistema de inventario está entre 216 y 224. Se alcanza cuando el inventario máximo es de aproximadamente 100 unidades y el punto de reorden alrededor de 40.

### 3.3 Análisis de parada de la simulación

Una simulación de terminación tiene un evento natural que especifica la longitud de cada ejecución. Esta simulación está "terminando" porque estamos interesados en la salida del sistema después de 120 unidades de tiempo, pensando en las unidades de tiempo como meses, para simular los eventos durante 10 años. Una simulación sin terminación no tiene tal evento final y el análisis se centra en el rendimiento a largo plazo del sistema, pero este no es caso para nuestro análisis.

## 4 Modelo Matemático

### 4.1 Descripción del modelo de simulación

Estamos considerando una tienda que almacena un tipo particular de producto. Los clientes que demandan este producto se presentan de acuerdo a un proceso de Poisson con tasa  $\lambda$ , y la cantidad demandada por cada uno es una variable aleatoria que tiene distribución G. Con el fin de satisfacer las demandas, el tendero debe mantener un cantidad de producto disponible, y siempre que el inventario disponible sea bajo, Las unidades adicionales se piden al distribuidor. El tendero utiliza una política  $(s, S)$  de pedidos; es decir, siempre que el inventario disponible sea inferior a  $s$  y no hay una orden pendiente en ese momento, entonces se ordena una cantidad para traerla a colación al  $S$ , donde  $s < S$ . Es decir, si el nivel de inventario actual es  $x$  y no hay ningún pedido pendiente, entonces si  $x < s$  se ordena la cantidad  $S - x$ . El costo de ordenar y unidades del producto es una función especificada  $c(y)$ , y se necesitan  $L$  unidades de tiempo hasta que el pedido se entrega, y el pago se realiza en el momento de la entrega. Además la tienda paga un costo de mantenimiento de inventario de  $H$  por producto por unidad de tiempo. Se supone además, cada vez que un cliente exija más del producto de lo que hay actualmente disponible, entonces la cantidad disponible se vende y el resto del pedido se pierde para la tienda, representando pérdida.

#### 4.1.1 Definición de variables

- Variables de tiempo:
  - $t$  tiempo general
  - $t_A$  tiempo de llegada de clientes
  - $t_M$  tiempo de llegada de orden
  - $t_H$  tiempo de pago por almacenaje
- Variables contadoras:
  - $C$  costo por reposición de productos
  - $H$  costo por almacenaje
  - $P$  pérdida por insatisfacción de la demanda del cliente
- Variables de estado:
  - $x$  cantidad de productos en inventario
  - $y$  cantidad de productos a reponer

#### 4.1.2 Inicialización de variables

- $t = C = H = P = y = 0$
- $x$  cantidad inicial en inventario
- Generar  $t_0$  y  $t_A = t_0$ ,  $t_M = \infty$

#### 4.1.3 Definición de eventos

- Eventos de llegada de clientes:
  - $t = t_A$ , actualizar línea de tiempo
  - Generar  $X$ ,  $w = \min(x, X)$ , se genera solución factible
  - $P = P + r(X - w)$ , actualizar posibles pérdidas
  - $x = x - w$ , actualizar inventario
  - Si  $x < s$  y  $y = S - x$  entonces  $y = 1$  y  $t_M = t + L$ , determinar si se debe colocar orden de compra
- Eventos de llegada de orden:
  - $t = t_<$ , actualizar línea de tiempo
  - $C = C + c(y)$ , actualizar gastos por reposición
  - $x = x + y$ , actualizar inventario
  - $y = 0$ ,  $t_M = \infty$ , no hay orden pendiente
- Evento de pago por almacenaje:
  - $H = H + xh$ , actualizar costo de almacenaje
  - $t = t_H$ , actualizar línea de tiempo
  - $t_H = t + 1$ , generar próximo pago por almacenaje

#### 4.2 Supuestos y restricciones

- La cantidad de productos demandados por cada cliente sigue una distribución uniforme.
- El tiempo entre llegadas de clientes sigue una distribución exponencial con tasa  $\lambda$ .
- El costo de pedido y unidades del producto sigue una función  $c(y)$ .
- El costo de mantenimiento de inventario por unidad de tiempo es  $h$ .
- La cantidad máxima de productos en el inventario es  $S$ .
- El punto de reorden es  $s$ .
- Si un cliente no puede satisfacer por completa su demanda, vacía el inventario
- Si dos eventos ocurren en el mismo momento, se procesan en el siguiente orden: pago por almacenaje, llegada de orden, llegada de clientes .