Pontificia Universidad Católica de Chile

Fundamentos de simulación – ICS2133

Profesor: Pedro Gazmuri

Informe 2

Integrantes:

- Patricio Aguilera

- Trinidad Aguilera

- Jimmy Gallardo

Fecha: 24 de octubre de 2023

Tabla de contenidos

[Resumen ejecutivo 2](#_Toc1569834350)

[Descripción general del supermercado 3](#_Toc22992855)

[Modelo Conceptual 4](#_Toc1662092980)

[Descripción 4](#_Toc1420557081)

[Supuestos 5](#_Toc1917745066)

[Objetivos y medidas de desempeño 5](#_Toc1752253315)

[Variables de estado 6](#_Toc771692709)

[Tipos de eventos 6](#_Toc1537731208)

[Toma de datos y análisis 7](#_Toc211310429)

[Datos recolectados 7](#_Toc1222392069)

[Resultados, análisis y validación 8](#_Toc1840306469)

[Tiempo de espera 8](#_Toc1179533374)

[Porcentaje de ocupación 9](#_Toc144745233)

[Intervalo de confianza 11](#_Toc747034932)

[Porcentaje de error 12](#_Toc1363365002)

[Conclusiones 12](#_Toc190797039)

[Codigo 13](#_Toc885146397)

[Main 14](#_Toc1805891000)

[Simulador 14](#_Toc1741048215)

[Simulador 17](#_Toc1269905570)

[Supermercado 18](#_Toc333151159)

[Caja 22](#_Toc891304307)

[Resultados 22](#_Toc1703986590)

[Parametros 23](#_Toc1808101045)

[No homogeneo 24](#_Toc1822240043)

[Intervalos 25](#_Toc546945519)

[Guardar datos 27](#_Toc414186672)

[Excel 27](#_Toc613844804)

[Python 29](#_Toc379166134)

[Anexos 29](#_Toc966518350)

# Resumen ejecutivo

En este documento, como un grupo del curso "Fundamentos de Simulación" de la carrera de Ingeniería en la Pontificia Universidad Católica de Chile, presentaremos el estudio que hemos llevado a cabo en el supermercado Jumbo, situado en el Costanera Center. El propósito de este estudio es observar, analizar y modelar el comportamiento de este sistema con el objetivo de implementar mejoras que impacten en aspectos como los tiempos de atención, la longitud de las filas y/o la disponibilidad de cajas en diferentes horarios.

Nuestra aproximación a este objetivo se llevará a cabo de manera gradual, y en particular, para esta entrega, nos enfocamos en la creación de un modelo funcional mínimo que reproduzca de manera razonablemente precisa el sistema que hemos seleccionado. Para lograr esto, hemos realizado mediciones de diversas tasas de operación del supermercado, incluyendo los tiempos de llegada de los clientes, los tiempos de compra y los tiempos de servicio en las cajas normales y rápidas. A partir de estas mediciones, hemos calculado las tasas de ocurrencia de cada variable de input. En relación a las llegadas, estas fueron consideradas como un Proceso de Poisson No Homogéneo. Por otra parte, los tiempos de compra y de atención en las cajas, se consideró que distribuyen Exponencial y se ajustaron sus parámetros dados los datos recolectados.

Posteriormente, se desarrolló la simulación del supermercado, para el cual se consideraron tanto las características técnicas del lugar como también los distintos procesos que ocurren dentro del mismo. Todo esto con el fin de poder comparar los resultados computacionales con los obtenidos en las mediciones empíricas. Todo este proceso de simulación nos permitirá analizar los resultados y evaluar su precisión. Para posteriormente realizar implantar y/o proponer alguna mejora que computacionalmente muestre mejoras en las distintas medidas de desempeño seleccionados.

En relación a los resultados obtenidos, por un lado, respecto al tiempo de espera promedio en las colas se obtuvo un valor cercano a 1.5 minutos para las cajas de tipo normal y las de auto servicio. Por otro lado, respecto al porcentaje de ocupación, para las cajas de tipo normal se obtuvo un valor cercano al 35%, mientras que para las cajas de tipo rápida se obtuvo un valor de un 92% de ocupación, aproximadamente. Estos valores fueron validados mediante los intervalos de confianza y sus errores relativos.

Finalmente, como propuesta futura se quiere hacer un análisis de sensibilidad respecto a la cantidad de cajas de tipo normal que se encuentran abiertas, ya que para esta entrega se asumió que todas las cajas estaban disponibles. Sin embargo, en la realidad esta cantidad puede variar durante los distintos momentos de tiempos de la jornada.

# Descripción general del supermercado

El supermercado Jumbo ubicado dentro del Mall Costanera Center es el que fue escogido para hacer la toma de datos. La dirección es Andrés Bello 2425, Piso 1, Providencia, Región Metropolitana.

El tamaño del supermercado es de 15000 . Respecto a las cajas de servicio, La cantidad es 33 de servicio normal y 12 de servicio rápido.

La comuna de Providencia, de la región Metropolitana, es una de las mejores calificadas a nivel socioeconómico. De acuerdo a estudios, el nivel de prioridad social es muy bajo en comparación con otras comunas, esto quiere decir que debido a su buen nivel socioeconómico es de las comunas que no requieren una intervención para mejorar las condiciones de vida. El resultado de este indicador está basado en las distintas mediciones hechas a la comuna. Un ejemplo de estos valores es el ingreso, y el porcentaje de personas en el tramo del 40% son 21.7, muy bajo en comparación con la comuna de la Pintana, donde este índice es de 95.5.

# Modelo Conceptual

## Descripción

Las mediciones que tomamos como equipo las hicimos en distintas fechas, el 26 y 27 de agosto fueron las primeras fechas en las que tomamos datos. Las otras fueron tomadas el 30 de septiembre, 3, 10 y 14 de octubre. Estas últimas fueron tomadas en distintos horarios. Dicho esto, nuestra propuesta de levantamiento de información o, mejor dicho, las estrategias que utilizamos para recapitular los datos que necesitamos fueron tres. A continuación, se mencionan cada una de ellas de manera detallada.

En primer lugar, medimos los tiempos de atención tanto en cajas normales como en cajas de atención rápida. Para esto, nos repartimos en distintas cajas del lugar y registramos los tiempos de atención de clientes en estas cajas con ayuda de un cronómetro. El objetivo fue tomar una gran variedad de cajas para disminuir la correlación cajero-caja, ya que algunos cajeros pueden tener más experiencia de atención y, por ende, realizan su trabajo de manera más rápida. Paralelamente, registramos los tiempos de atención de los clientes en las cajas rápidas de la misma manera que en las cajas normales.

En segundo lugar, consideramos la cantidad de personas que ingresa a al recinto en un periodo determinado de tiempo o, dicho de otro modo, la tasa de llegada de clientes. Esta se calculó en periodos de una hora y se contaron cuántas personas ingresaron el recinto en dicho tiempo. Este cálculo se realizará en varias ocasiones para así asegurar un número relativamente cercano a la realidad. Cabe destacar que se consideraba como un cliente cuando llegaba un grupo de personas que venían a realizar juntos la compra. Para no tener complicaciones, se tomó la decisión que la llegada de cada cliente se registrará en una calculadora. Además, hay que considerar que la llegada de clientes debe modelarse como un proceso de Poisson No Homogéneo, por lo que era necesario realizar varias mediciones en intervalos de 60 minutos.

Por último, registramos los tiempos de duración de las compras de los distintos clientes de dos maneras diferentes. La primera forma consistió en hacerle un seguimiento a los diferentes clientes que iban ingresando, seleccionándolos de manera aleatoria y considerando distintos grupos, tales como familias, parejas, personas solas, adultos mayores, jóvenes, etc., para lograr que nuestros valores muestren una gran variedad de opciones de compra. La segunda opción que realizamos tomó origen en no invadir a las personas y consistió en crear diferentes listas de compras realistas para luego simular dichas compras dentro del supermercado.

## Supuestos

Dado que nosotros recrearemos este sistema (supermercado) y es difícil recrear al 100% lo que sucede en la vida real, es necesario considerar unos supuestos iniciales. A continuación, se presentan los supuestos que consideramos adecuados para la modelación de nuestro problema, dada la experiencia que tuvimos durante la medición de datos:

* Capacidad ilimitada del supermercado: dado que en ningún caso histórico hemos visto que nieguen el acceso a un supermercado debido a su capacidad de atención, es un supuesto valido a la hora de realizar la programación.
* Un cliente corresponde a un grupo que realiza la misma compra: nosotros no consideraremos la cantidad total de personas que entran al local, sino que consideramos grupos de personas que ingresan. Por ejemplo, si entra una familia de 5 integrantes, esta fue considerada un cliente ya que, si era considerada como grupo, después no había una forma para identificar a los clientes y asignarlos a un grupo respectivo.
* Toda persona que ingrese al recinto si o si realizará algún tipo de compra: Dado que no podemos hacer un seguimiento a cada uno de los clientes que ingresan al supermercado para verificar que realicen una compra, consideramos que era prudente generalizar que todos los clientes no se iban del supermercado sin haber comprado algo antess
* Medición del tiempo de atención en la caja: la medición empezaba desde que el cliente comenzaba a dejar sus productos encima del mostrador hasta que recibía la boleta. No consideramos el tiempo en que tardaba en recoger sus artículos.
* Una vez que un cliente haya elegido la cola de la caja en la que va a pagar, este no se cambia a otra después.
* Un cliente escoge la caja que tiene la cola más corta.
* El cliente no escoge las cajas de servicio rápido si es que lleva muchas compras, ejemplo de esto sería un carro lleno.
* Si una caja tiene un problema técnico, se contabilizará ese tiempo de demora.
* Asumimos que que todas las cajas del supermercado se encuentran abierta, ya que no pudimos hacer un análisis entre los diferentes momentos del día de la cantidad de cajas abiertas. Eventualmente, podemos realizar un análisis de sensibilidad.
* La elección del tipo de caja a la que cada cliente será atendido se determina mediante una sencilla probabilidad que considera la cantidad de cajas totales de cada tipo.

## Objetivos y medidas de desempeño

Dado lo anterior, como grupo tenemos como principal objetivo analizar el funcionamiento de esta institución durante un día de trabajo y hemos decidido analizar las mediciones de distintos fines de semana. El trabajo principal que se realizará durante este periodo será medir los distintos indicadores de desempeño (KPI’ s). A continuación, se muestran las medidas de desempeño que planeamos medir y analizar:

* Tiempo promedio de espera en la cola: este corresponde al tiempo promedio que un cliente pasa esperando en la cola antes de ser atendido en la caja que escogió. Un tiempo de espera largo puede indicar problemas en la eficiencia de atención al cliente.

* Porcentaje de ocupación promedio en cada caja: con esta medida de desempeño podemos obtener el porcentaje promedio de tiempo durante el cual las cajas estuvieron ocupadas. Una baja utilización podría sugerir que hay demasiadas cajas para la tasa de llegada de clientes.

* Número máximo de clientes en una cola: esta medida de desempeño identifica el punto en el que se produce la mayor congestión en una cola. Puede ayudar a evaluar si se necesita agregar más cajas o redistribuir a los cajeros de manera más eficiente.

* Número promedio de clientes en el supermercado: nos indica cuántos clientes, en promedio, se encuentran dentro del supermercado en cualquier momento dado. Puede ayudar a determinar si el supermercado está operando en su capacidad máxima. Para este caso en particular, se consideró que la capacidad era ilimitada.

## Variables de estado

* Estado de las cajas: Con esto verificamos en el estado de ocupación de cada una de las cajas: libre u ocupada.

* Número de clientes en el supermercado: Con esto hicimos un conteo de cuantos clientes hay en el supermercado en un tiempo determinado. Este aumentará a medida que ingresen nuevos usuarios y disminuirá cuando cada uno termine de ser atendido en la caja.

* Número de clientes en cada cola de la caja: A partir del uso de esta variable contabilizábamos el número de clientes que deberían estar esperando en cada cola. Con eso, cada cliente escogía la cola más corta para ser atendido.
* Tasa de llegada de clientes: La tasa de llegada la cual mide cuánto tiempo llegan los clientes en promedio.

## Tipos de eventos

Los eventos son puntos en el tiempo en los que ocurren cambios significativos en el estado del sistema. Cada uno de los eventos tiene un impacto en las variables de estado y avanza la simulación en el tiempo. Para la simulación del supermercado, podemos definir los siguientes eventos:

* Llegada de un cliente: este evento ocurre cuando llega un nuevo cliente al supermercado. En particular, esto ocurre de acuerdo a un proceso de Poisson No Homogéneo.

* Selección de cola: cuando un cliente ha terminado de comprar, este se va la caja cuya cola es la más corta. Este evento ocurre cuando un cliente elige una cola específica para esperar su turno en la caja.

* Inicio de atención: ocurre cuando un cliente llega a una caja y la caja está libre, por lo que comienza el servicio.

* Término de atención: ocurre cuando un cliente ha sido completamente atendido en la caja y se va del supermercado (sistema).

* Término de la simulación: Este evento ocurre cuando el reloj de la simulación registra el término de la jornada del supermercado. Posteriormente, se vuelve a realizar otra simulación y así sucesivamente hasta que hayan finalizado las N réplicas.

# Toma de datos y análisis

## Datos recolectados

A continuación se encuentran las tablas con los datos recolectados dentro del supermercado.

|  |  |  |
| --- | --- | --- |
| Hora de Inicio | Hora de término | Tasa de llegada (personas/minuto) |
| 8 | 9 | 4.46 |
| 9 | 10 | 9.3 |
| 10 | 11 | 13.36 |
| 11 | 12 | 14.03 |
| 12 | 13 | 15.9 |
| 13 | 14 | 17.03 |
| 14 | 15 | 17.5 |
| 15 | 16 | 18.1 |
| 16 | 17 | 17.23 |
| 17 | 18 | 16.2 |
| 18 | 19 | 16.8 |
| 19 | 20 | 17.7 |
| 20 | 21 | 16.6 |

Tabla 1: Tasas de llegada de las personas

Los datos obtenidos para el tiempo de compra y atención en las cajas se encuentran en la sección de Anexos. A continuación, se muestran las tasas, considerando que se ajustaron los datos a una distribución Exponencial.

|  |  |  |
| --- | --- | --- |
| Tasa de compra | Tasa de atención caja normal | Tasa de atención caja rápida |
| 22.59 | 1.56 | 1.49 |

Tabla 2: Tasas de atención y compra

A partir de los valores observador se ve que los tiempos de atención y compra son bastante certeros si es que lo comparamos con la realidad.

|  |  |  |  |
| --- | --- | --- | --- |
| PARÁMETROS | | | |
| HORA INICIO | HORA CIERRE | CANTIDAD CAJAS NORMALES | CANTIDAD CAJAS RAPIDAS |
| 9:00 AM | 9:00 PM | 33 | 12 |

Tabla 3: Parámetros del supermercado

# Resultados, análisis y validación

Hay que considerar que las cajas con ID entre 0 y 32, inclusive, corresponden a las cajas normales, mientras que el resto corresponde a las cajas de atención rápida, es decir, las que son auto servicio.

## Tiempo de espera

Este corresponde al tiempo promedio que un cliente pasa esperando en la cola antes de ser atendido en la caja que escogió. Un tiempo de espera largo puede indicar problemas en la eficiencia de atención al cliente. A continuación, se muestran los resultados del tiempo promedio de espera en cola de cada caja para la réplica 0 a modo de ejemplo.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Espera réplicas simulación | | | | | | |
| Réplica | Caja | Valor [min] |  | Réplica | Caja | Valor [min] |
| 0 | 0 | 1,529633 | 0 | 23 | 1,61079 |
| 0 | 1 | 1,51109 | 0 | 24 | 1,44916 |
| 0 | 2 | 1,821753 | 0 | 25 | 1,536193 |
| 0 | 3 | 1,571503 | 0 | 26 | 1,507238 |
| 0 | 4 | 1,682254 | 0 | 27 | 1,689029 |
| 0 | 5 | 1,488786 | 0 | 28 | 1,568524 |
| 0 | 6 | 1,693295 | 0 | 29 | 1,69309 |
| 0 | 7 | 1,574315 | 0 | 30 | 1,397777 |
| 0 | 8 | 1,712233 | 0 | 31 | 1,451898 |
| 0 | 9 | 1,418008 | 0 | 32 | 1,634157 |
| 0 | 10 | 1,589316 | 0 | 33 | 1,457801 |
| 0 | 11 | 1,618681 | 0 | 34 | 1,642089 |
| 0 | 12 | 1,594842 | 0 | 35 | 1,502602 |
| 0 | 13 | 1,398409 | 0 | 36 | 1,389429 |
| 0 | 14 | 1,582181 | 0 | 37 | 1,425472 |
| 0 | 15 | 1,647506 | 0 | 38 | 1,362703 |
| 0 | 16 | 1,684998 | 0 | 39 | 1,541524 |
| 0 | 17 | 1,655576 | 0 | 40 | 1,447444 |
| 0 | 18 | 1,556595 | 0 | 41 | 1,607385 |
| 0 | 19 | 1,424105 | 0 | 42 | 1,462094 |
| 0 | 20 | 1,530184 | 0 | 43 | 1,415514 |
| 0 | 21 | 1,574941 | 0 | 44 | 1,413125 |
| 0 | 22 | 1,702729 | ... | ... | ... |

Tabla 4: Tiempos de espera promedio para cada réplica

De los resultados obtenidos, podemos notar que en promedio hay apróximadamente una persona antes de que un cliente es atendido, considerando que el tiempo de atención distribuye exponencial con tasa 1.5635 y 1.488 para las cajas de tipo normal y rápida, respectivamente. Respecto a la magnitud, la respuesta es bastante relativa porque depende si el cliente estará dispuesto a esperar dicho tiempo. No obstante, dada la experiencia que hemos tenido en los supermercados (y cualquier otra persona), se podría decir que los resultados obtenidos son bastante prudentes. Sin embargo, hay que tener en cuenta que en la simulación no se considera ningún tipo de caso especial, por ejemplo, que una cajera vaya al baño, haya un problema con la lectura de un código del producto, falle el sistema de Transbank, entre otras.

El siguiente paso consistió en calcular el tiempo promedio de espera en cola entre todas las cajas y réplicas de la simulación. Adicionalmente, se calculó la desviación estándar ya que, este valor era necesario para el cálculo del intervalo de confianza y su error relativo. Se hizo la distinción entre cajas rápidas y normales. Los resultados obtenidos fueron los siguientes:

|  |  |  |
| --- | --- | --- |
| Tipo de Caja | Tiempo espera [min] Promedio | Dev. Estandar Espera Promedio |
| Normal | 1.55 | 0.12 |
| Rápida | 1.46 | 0.07 |

Tabla 5: Tiempos promedios de cajas y desv. Estándar

## Porcentaje de ocupación

Con esta medida de desempeño podemos obtener el porcentaje promedio de tiempo durante el cual las cajas estuvieron ocupadas. Una baja utilización podría sugerir que hay demasiadas cajas para la tasa de llegada de clientes. A continuación, se muestran los resultados del porcentaje de ocupación promedio de cada caja para la réplica 0 a modo de ejemplo.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ocupación réplicas simulación | | | | | | |
| Réplica | Caja | Valor [%] |  | Réplica | Caja | Valor [%] |
| 0 | 0 | 35,92021792 | 0 | 23 | 33,91548354 |
| 0 | 1 | 34,71172981 | 0 | 24 | 33,19604946 |
| 0 | 2 | 39,45294469 | 0 | 25 | 35,52183224 |
| 0 | 3 | 37,5976746 | 0 | 26 | 35,00094038 |
| 0 | 4 | 38,01684036 | 0 | 27 | 37,27562167 |
| 0 | 5 | 34,48221125 | 0 | 28 | 36,26958178 |
| 0 | 6 | 39,75178195 | 0 | 29 | 37,76941483 |
| 0 | 7 | 37,86124715 | 0 | 30 | 34,04913067 |
| 0 | 8 | 39,11468917 | 0 | 31 | 35,84145083 |
| 0 | 9 | 33,87769283 | 0 | 32 | 37,31708414 |
| 0 | 10 | 37,99966495 | 0 | 33 | 93,2911767 |
| 0 | 11 | 38,43501857 | 0 | 34 | 94,17887461 |
| 0 | 12 | 37,68262416 | 0 | 35 | 94,62890775 |
| 0 | 13 | 33,26272476 | 0 | 36 | 92,12994012 |
| 0 | 14 | 33,07986533 | 0 | 37 | 92,07858943 |
| 0 | 15 | 37,18353955 | 0 | 38 | 95,86445815 |
| 0 | 16 | 35,43009324 | 0 | 39 | 93,48517287 |
| 0 | 17 | 37,66398729 | 0 | 40 | 92,42938202 |
| 0 | 18 | 36,16456268 | 0 | 41 | 93,61176232 |
| 0 | 19 | 34,46122926 | 0 | 42 | 91,73867347 |
| 0 | 20 | 38,11839846 | 0 | 43 | 91,04058732 |
| 0 | 21 | 37,0963137 | 0 | 44 | 91,49724182 |
| 0 | 22 | 37,05930349 | ... | ... | ... |

Tabla 6: Porcentajes de ocupación de cajas de cada réplica

Sobre los resultados obtenidos para los porcentajes promedio de ocupación, ocurre algo similar que en los resultados de los tiempos de espera en las colas de las cajas. Dado que estamos trabajando con una simulación, el modelo no considera fallas en el sistema o tiempos espaciales, tales como falla en los equipos, horas de almuerzo, turnos de las cajeras, etc. Tomando lo anterior en consideración, podemos determinar que, en promedio, las cajas normales no se encuentran saturadas y su porcentaje de ocupación es apróximadamente un 35%, mientras que las cajas rápidas se encuentran la mayor parte de la jornada ocupadas. Cabe destacar que para este informe y la respectiva simulación se consideró que todas las cajas se encuentran abiertas. Eventialmente, podemos realizar un análisis de sensibilidad para las cajas normales relacionado con la cantidad que se encuentran abiertas.

El siguiente paso consistió en calcular el porcentaje promedio de ocupación entre todas las cajas y réplicas de la simulación. Adicionalmente, se calculó la desviación estándar ya que, este valor era necesario para el cálculo del intervalo de confianza. Se hizo la distinción entre cajas normales y rápidas, ya que se comportan de manera diferente. Los resultados obtenidos fueron los siguientes:

|  |  |  |
| --- | --- | --- |
| Tipo de caja | Porcentaje de Ocupación [%] Promedio | Dev. Estandar Ocupación Promedio |
| Normal | 35.71 | 2.28 |
| Rápida | 92.71 | 1.61 |

Tabla 7: Porcentaje de ocupación promedio y Desv. Estándar

## Intervalo de confianza

A continuación, se presentan los intervalos de confianza y para un nivel de confianza del 90% y 95%, respectivamente, para el tiempo de espera promedio en ambos tipos de cajas y su respectivo porcentaje de ocupación.

|  |  |  |  |
| --- | --- | --- | --- |
| INTERVALO DE CONFIANZA | | | |
| CONFIANZA (%) |  | TIPO DE CAJA | |
| NORMAL | RÁPIDA |
| 90 | ESPERA | [1.51 ; 1,59] | [1.44 ; 1.48 ] |
| OCUPACIÓN | [35 ; 36.41] | [92.21 ; 93.21] |
| 95 | ESPERA | [1.51 ; 1.6] | [1.44 ; 1.49] |
| OCUPACIÓN | [34.86 ; 36.56] | [92.11 ; 93.31] |

Tabla 8: Intervalo de confianza para tiempos de espera y porcentaje de ocupación

Para calcular el intervalo de confianza con un nivel de significancia α, utilizamos la siguiente fórmula:



Antes de hacer el análisis, debemos tomar en cuenta que en estadística el nivel de confianza es una medida que se utiliza para expresar la confiabilidad o certeza que se tiene en los resultados de nuestro estudio. En otras palabras, representa el porcentaje de veces que se espera que un intervalo de confianza contenga el parámetro de interés. Por lo tanto, un nivel de confianza más alto implica un intervalo de confianza más amplio, ya que necesita ser más conservador en la estimación.

Dicho esto, a partir de los resultados obtenidos, podemos notar que el intervalo para un nivel de confianza de 95% es mayor en comparación al intervalo a un 90% de confianza para cada tipo de cajas y su respectivo tiempo de espera y porcentaje de ocupación.

En relación con el análisis, podemos determinar que nuestros resultados son bastante fidedignos, ya que, para cada tipo de caja, cada nivel de confianza y considerando el tiempo de espera y los porcentajes de ocupación de las cajas, podemos observar que los promedios de nuestras muestras se encuentran dentro de su respectivo intervalo de confianza. Esto significa que, con el nivel de confianza establecido, la estimación de nuestra media poblacional es consistente con nuestra muestra. En otras palabras, el intervalo de confianza propociona evidencia de que cada una de las medias reales de la población podrían estar cerca del valor que calculamos a partir de cada una de nuestras muestras

Por ejemplo, considerando el tiempo de espera promedio para las cajas de tipo normal y un nivel de confianza de un 90%, podemos notar que nuestra media muestral, cuyo valor es 1.55 min, se encuentra dentro del intervalo de confianza obtenido, el cual corresponde a [1.51; 1.59].

## Porcentaje de error

La tabla presentada a continuación muestra el porcentaje de error del de los intervalos de confianza que fueron planteado anteriormente. Esta medición está hecha tanto para el 90% y 95%.

|  |  |  |  |
| --- | --- | --- | --- |
| ERROR | | | |
| CONFIANZA (%) |  | TIPO DE CAJA | |
| NORMAL | RAPIDA |
| 90 | ESPERA | 2.47 | 1.48 |
| OCUPACION | 1.97 | 0.54 |
| 95 | ESPERA | 2.98 | 1.78 |
| OCUPACION | 2.38 | 0.65 |

Tabla 9: Porcentaje de error del tiempo de espera y porcentaje de ocupacipon de cada caja

Por otra parte, para calcular el porcentaje de error de cada intervalo, utilizamos la siguiente fórmula:



Donde X(n) corresponde al promedio entre las réplicas del tiempo promedio de espera o el porcentaje de ocupación.

Dicho lo anterior, es coherente que el porcentaje de error para un nivel de confianza de 95% sea mayor en comparación al nivel de confianza de 90% ya que, tal como dijimos anteriormente, mientras mayor nivel de confianza, el intervalo es más amplio y, por lo tanto, la mitad del largo de dicho intervalo también lo será. Adicionalmente, el promedio entre réplicas del tiempo promedio de espera en cola para cada una de las cajas es el mismo para ambos casos, por lo tanto, para un nivel de confianza de 95% estamos dividiendo un número mayor por el mismo promedio, lo cual provoca un mayor porcentaje de error.

En otras palabras, podemos notar que nuestros errores relativos de los intervalos de confianza obtenidos son bastante pequeños, lo cual nos indica que nuestra estimación es más precisa y de mayor confianza en la posición de la media.

# Conclusiones

A partir de la simulación realizada del supermercado Jumbo ubicado en el mall Costanera Center, podemos concluir que este no se encuentra saturado. Respecto a los tiempo de espera en las colas, este corresponde al tiempo promedio que un cliente pasa esperando en la cola antes de ser atendido en la caja que escogió. Por un lado, para las cajas rápidas se obtiene que, en promedio, un cliente debe esperar 1.46 minutos antes de ser atendido. Por otro lado, para las cajas de tipo normal se obtiene un valor de 1.55 minutos. Respecto al porcentaje de ocupación, este corresponde al porcentaje promedio de tiempo durante el cual las cajas estuvieron ocupadas durante la jornada. En relación a las cajas rápidas, se obtuvo un valor cercano al 92% de esta medida de desempeño, mientras que para las cajas de tipo normal se obtuvo un valo de aproximadamente de un 35%. Esta diferencia se puede explicar por diferentes razones. Una de ellas es que la cantidad de cajas rápidas es menos de la mitad de la cantidad de cajas de tipo normal. Adicionalmente, para esta entrega se consideró que todas las cajas se encontraban abiertas. Por esta razón, se plantea que para la próxima entrega se podría realizar un análisis de sensibilidad en la cantidad de cajas abiertas durante la jornada y evaluar cómo se ven afectadas las medidas de desempeño utilizadas.

Respecto a la validación de nuestros resultados obtenidos, se calcularon los intervalos de confianza y sus errores relativos. Estos revelan la confiabilidad de las estimaciones realizadas dados los niveles de confianza utilizados. En otras palabras, se puede decir que las estimaciones son precisas y confiables.

En síntesis, la investigación análisis de los datos recopilados ayudan a comprender el funcionamiento del supermercado. Gracias a esto, podría aumentar la eficiencia de la gestión realizada y así hacer mejorar la experiencia de los clientes.

# Codigo

## Main

from simulador import Simulador

print("inicio programa")

if \_\_name\_\_ == "\_\_main\_\_":

simulador = Simulador()

simulador.simular\_replica()

print("fin programa")

Simulador

import parametros as p

from simulacion import Simulacion

import time

from guardar\_datos import guardar\_resultados, crear\_archivo, cargar\_datos

from intervalos import Intervalo

import math

class Simulador:

def \_\_init\_\_(self):

self.dict\_duracion\_replica = {}

crear\_archivo()

self.dict\_espera = {i: 0 for i in range(p.CANTIDAD\_CAJAS\_NORMALES + p.CANTIDAD\_CAJAS\_RAPIDAS)}

self.dict\_ocupacion = {i: 0 for i in range(p.CANTIDAD\_CAJAS\_NORMALES

+ p.CANTIDAD\_CAJAS\_RAPIDAS)}

self.espera = {}

self.ocupacion = {}

self.sd\_espera = {}

self.sd\_ocupacion = {}

self.int\_espera\_90 = {}

self.int\_ocupacion\_90 = {}

self.int\_espera\_95 = {}

self.int\_ocupacion\_95 = {}

self.dict\_espera\_simulacion = {}

self.dict\_ocupacion\_simulacion = {}

def simular\_replica(self):

for replica in range(p.REPLICAS):

self.simulacion = Simulacion()

tiempo\_inicio = time.time()

self.simulacion.simular()

self.sumar\_tiempos\_espera\_cola()

self.sumar\_porcentaje\_ocupacion()

dict\_arreglado\_espera = self.arreglar\_dict(self.simulacion.dict\_promedio\_tiempo\_espera\_cola)

dict\_arreglado\_ocupacion = self.arreglar\_dict(self.simulacion.dict\_promedio\_ocupacion)

self.dict\_espera\_simulacion[replica] = dict\_arreglado\_espera

self.dict\_ocupacion\_simulacion[replica] = dict\_arreglado\_ocupacion

tiempo\_fin = time.time()

duracion\_replica = tiempo\_fin - tiempo\_inicio

self.dict\_duracion\_replica[replica] = duracion\_replica

self.espera = self.calcular\_espera\_promedio()

self.ocupacion = self.calcular\_ocupacion\_promedio()

self.sd\_espera = self.calcular\_sd\_espera()

self.sd\_ocupacion = self.calcular\_sd\_ocupacion()

guardar\_resultados("datos\espera\_replicas.json", self.dict\_espera\_simulacion)

guardar\_resultados("datos\ocupacion\_replicas.json", self.dict\_ocupacion\_simulacion)

guardar\_resultados("datos\dict\_tiempo\_de\_ejecucion.json", self.dict\_duracion\_replica)

guardar\_resultados("datos\dict\_tiempo\_promedio\_espera\_cola.json", self.espera)

guardar\_resultados("datos\dict\_porcentaje\_ocupacion\_cajas.json", self.ocupacion)

guardar\_resultados("datos\sd\_espera.json", self.sd\_espera)

guardar\_resultados("datos\sd\_ocupacion.json", self.sd\_ocupacion)

self.intervalo = Intervalo()

self.int\_espera\_90 = self.intervalo.calcular\_intervalo\_espera(90)

self.error\_90\_espera = self.intervalo.error\_90\_espera

self.int\_ocupacion\_90 = self.intervalo.calcular\_intervalo\_ocupacion(90)

self.error\_90\_ocupacion = self.intervalo.error\_90\_ocupacion

self.int\_espera\_95 = self.intervalo.calcular\_intervalo\_espera(95)

self.error\_95\_espera = self.intervalo.error\_95\_espera

self.int\_ocupacion\_95 = self.intervalo.calcular\_intervalo\_ocupacion(95)

self.error\_95\_ocupacion = self.intervalo.error\_95\_ocupacion

guardar\_resultados("datos\int\_90\_espera.json", self.int\_espera\_90)

guardar\_resultados("datos\int\_95\_espera.json", self.int\_espera\_95)

guardar\_resultados("datos\int\_90\_ocupacion.json", self.int\_ocupacion\_90)

guardar\_resultados("datos\int\_95\_ocupacion.json", self.int\_ocupacion\_95)

guardar\_resultados("datos\error\_90\_espera.json", self.error\_90\_espera)

guardar\_resultados("datos\error\_95\_espera.json", self.error\_95\_espera)

guardar\_resultados("datos\error\_90\_ocupacion.json", self.error\_90\_ocupacion)

guardar\_resultados("datos\error\_95\_ocupacion.json", self.error\_95\_ocupacion)

def arreglar\_dict(self, dic):

dict\_ = {}

for tipo\_caja in dic.keys():

for nro in dic[tipo\_caja].keys():

valor = dic[tipo\_caja][nro]

dict\_[nro] = valor

return dict\_

def sumar\_tiempos\_espera\_cola(self):

for tipo\_caja in self.simulacion.dict\_promedio\_tiempo\_espera\_cola.keys():

for llave in self.simulacion.dict\_promedio\_ocupacion[tipo\_caja].keys():

self.dict\_espera[llave] += self.simulacion.dict\_promedio\_tiempo\_espera\_cola[tipo\_caja][llave]

def sumar\_porcentaje\_ocupacion(self):

for tipo\_caja in self.simulacion.dict\_promedio\_ocupacion.keys():

for llave in self.simulacion.dict\_promedio\_ocupacion[tipo\_caja].keys():

self.dict\_ocupacion[llave] += self.simulacion.dict\_promedio\_ocupacion[tipo\_caja][llave]

def calcular\_espera\_promedio(self):

dict\_ = {}

for llave in self.dict\_espera.keys():

dict\_[llave] = self.dict\_espera[llave] / p.REPLICAS

return dict\_

def calcular\_sd\_espera(self):

total = {i: 0 for i in range(p.CANTIDAD\_CAJAS\_NORMALES + p.CANTIDAD\_CAJAS\_RAPIDAS)}

for caja in self.espera.keys():

suma = 0

for replica in self.dict\_espera\_simulacion.keys():

dict\_espera = self.dict\_espera\_simulacion[replica]

for caja\_2 in dict\_espera.keys():

if caja == caja\_2:

resta = dict\_espera[caja\_2] - self.espera[caja]

cubo = resta \* resta

suma += cubo

total[caja] = math.sqrt(suma / (p.REPLICAS - 1))

return total

def calcular\_ocupacion\_promedio(self):

dict\_ = {}

for llave in self.dict\_ocupacion.keys():

dict\_[llave] = self.dict\_ocupacion[llave] / p.REPLICAS

return dict\_

def calcular\_sd\_ocupacion(self):

total = {i: 0 for i in range(p.CANTIDAD\_CAJAS\_NORMALES + p.CANTIDAD\_CAJAS\_RAPIDAS)}

for caja in self.ocupacion.keys():

suma = 0

for replica in self.dict\_ocupacion\_simulacion.keys():

dict\_ocupacion = self.dict\_ocupacion\_simulacion[replica]

for caja\_2 in dict\_ocupacion.keys():

if caja == caja\_2:

resta = dict\_ocupacion[caja\_2] - self.ocupacion[caja]

cubo = resta \* resta

suma += cubo

total[caja] = math.sqrt(suma / (p.REPLICAS - 1))

return total

## Simulador

import parametros as p

from supermercado import Supermercado

class Simulacion:

def \_\_init\_\_(self):

self.tiempo\_actual = 0

self.supermercado = Supermercado()

self.dict\_promedio\_tiempo\_espera\_cola = {"normal": {}, "rapida": {}}

self.dict\_promedio\_ocupacion = {"normal": {}, "rapida": {}}

def simular(self):

while self.supermercado.estado != "cerrado":

self.supermercado.actualizar\_tiempos(self.tiempo\_actual)

proximo\_evento, proximo\_tiempo = self.supermercado.proximo\_evento()

self.supermercado.ocurre\_evento(proximo\_evento, proximo\_tiempo, proximo\_tiempo + self.tiempo\_actual)

self.supermercado.actualizar\_dict\_tiempo\_anterior(proximo\_evento, proximo\_tiempo)

self.tiempo\_actual += proximo\_tiempo

if self.tiempo\_actual > p.JORNADA \* p.MINUTOS\_POR\_HORA:

self.supermercado.estado = "por\_cerrar"

if self.tiempo\_actual >= p.JORNADA \* p.MINUTOS\_POR\_HORA and \

self.supermercado.cantidad\_clientes\_comprando == 0 and \

self.supermercado.cantidad\_clientes\_caja == 0:

self.supermercado.estado = "cerrado"

self.tiempo\_promedio\_espera\_cola()

self.ocupacion\_promedio\_caja()

def tiempo\_promedio\_espera\_cola(self):

for tipo\_caja in self.supermercado.dict\_tiempo\_espera\_caja.keys():

for llave in self.supermercado.dict\_tiempo\_espera\_caja[tipo\_caja].keys():

suma = 0

largo = len(self.supermercado.dict\_tiempo\_espera\_caja[tipo\_caja][llave])

if largo == 0:

self.dict\_promedio\_tiempo\_espera\_cola[tipo\_caja][llave] = 0

else:

for tiempo in self.supermercado.dict\_tiempo\_espera\_caja[tipo\_caja][llave]:

suma += tiempo

promedio = suma / largo

self.dict\_promedio\_tiempo\_espera\_cola[tipo\_caja][llave] = promedio

return self.dict\_promedio\_tiempo\_espera\_cola

def ocupacion\_promedio\_caja(self):

for tipo\_caja in self.supermercado.dict\_ocupacion\_caja.keys():

for llave in self.supermercado.dict\_ocupacion\_caja[tipo\_caja].keys():

if self.supermercado.dict\_ocupacion\_caja[tipo\_caja][llave] == 0:

self.dict\_promedio\_ocupacion[tipo\_caja][llave] = 0

else:

self.dict\_promedio\_ocupacion[tipo\_caja][llave] = self.supermercado.dict\_ocupacion\_caja[tipo\_caja][llave] \* 100 / (p.JORNADA \* p.MINUTOS\_POR\_HORA)

return self.dict\_promedio\_ocupacion

Supermercado

import parametros as p

from caja import Caja

import numpy

from no\_homogeneo import tasa\_no\_homogeneo

import random

class Supermercado:

def \_\_init\_\_(self):

self.dict\_cajas = {"normal": {}, "rapida": {}}

self.tipo\_caja\_actual = None

self.dict\_tiempo\_proximo = {"llegada": p.INFINITO, "compra": p.INFINITO, "atencion": p.INFINITO}

self.dict\_evento\_anterior = {"evento": "", "tiempo": 0}

self.cantidad\_clientes\_comprando = 0

self.cantidad\_clientes\_caja = 0

self.estado = "abierto"

self.tiempos\_llegadas = []

self.tiempos\_fin\_compra = []

self.caja\_rapida = None

self.cantidad\_llegadas = 0

self.cantidad\_atenciones = 0

self.dict\_tiempo\_espera\_caja = {"normal": {}, "rapida": {}}

self.dict\_ocupacion\_caja = {"normal": {}, "rapida": {}}

self.instanciar\_cajas()

def instanciar\_cajas(self):

for c in range(p.CANTIDAD\_CAJAS\_NORMALES):

caja = Caja(c, "normal")

self.dict\_cajas["normal"][c] = caja

self.dict\_tiempo\_espera\_caja["normal"][c] = []

self.dict\_ocupacion\_caja["normal"][c] = 0

for cr in range(p.CANTIDAD\_CAJAS\_NORMALES, p.CANTIDAD\_CAJAS\_NORMALES + p.CANTIDAD\_CAJAS\_RAPIDAS):

caja = Caja(cr, "rapida")

self.dict\_cajas["rapida"][cr] = caja

self.dict\_tiempo\_espera\_caja["rapida"][cr] = []

self.dict\_ocupacion\_caja["rapida"][cr] = 0

def actualizar\_tiempos(self, tiempo\_actual):

eventos = self.verificar\_proximo\_evento()

for evento in eventos:

self.dict\_tiempo\_proximo[evento] = self.calcular\_tiempos(evento, tiempo\_actual)

def calcular\_tiempos(self, evento, tiempo\_actual):

if evento == "llegada":

tasa = tasa\_no\_homogeneo(tiempo\_actual)

return numpy.random.exponential(tasa)

elif evento == "compra":

tiempo = self.carrera\_exponencial\_compra()

return tiempo

elif evento == "atencion":

self.caja\_rapida, tiempo = self.caja\_mas\_rapida()

return tiempo

def carrera\_exponencial\_cajas(self):

dict\_id\_tiempo\_caja = {}

for tipo\_caja in self.dict\_cajas.keys():

for caja in self.dict\_cajas[tipo\_caja].values():

if caja.cola\_caja > 0:

tiempo = numpy.random.exponential(p.TASA\_ATENCION[tipo\_caja])

dict\_id\_tiempo\_caja[caja.id\_caja] = (tiempo, tipo\_caja)

return dict\_id\_tiempo\_caja

def carrera\_exponencial\_compra(self):

minimo = p.INFINITO

for \_ in range(self.cantidad\_clientes\_comprando):

tiempo = numpy.random.exponential(p.TASA\_COMPRA)

if tiempo < minimo:

minimo = tiempo

return minimo

def caja\_mas\_rapida(self):

dict\_ = self.carrera\_exponencial\_cajas()

minimo = p.INFINITO

llave\_oficial = p.INFINITO

for llave in dict\_.keys():

if dict\_[llave][0] < minimo:

minimo = dict\_[llave][0]

self.tipo\_caja\_actual = dict\_[llave][1]

llave\_oficial = llave

if llave\_oficial == p.INFINITO:

return None

return llave\_oficial, minimo

def elegir\_tipo\_caja(self, evento):

lista\_opciones\_valores = []

for key\_tipo\_caja in self.dict\_cajas.keys():

for caja in self.dict\_cajas[key\_tipo\_caja].values():

if caja.cola\_caja > 0 and key\_tipo\_caja not in lista\_opciones\_valores:

lista\_opciones\_valores.append(key\_tipo\_caja)

if self.tipo\_caja\_actual == None or evento == "compra":

tipo\_caja = random.choices(["rapida", "normal"], weights = [p.PROBABILIDAD\_ELEGIR\_CAJA\_RAPIDA, p.PROBABILIDAD\_ELEGIR\_CAJA\_NORMAL])

elif "normal" in lista\_opciones\_valores and "rapida" in lista\_opciones\_valores:

tipo\_caja = random.choices(["rapida", "normal"], weights = [p.PROBABILIDAD\_ELEGIR\_CAJA\_RAPIDA, p.PROBABILIDAD\_ELEGIR\_CAJA\_NORMAL])

elif "normal" in lista\_opciones\_valores and len(lista\_opciones\_valores) == 1:

tipo\_caja = random.choices(["rapida", "normal"], weights = [0, 1])

elif "rapida" in lista\_opciones\_valores and len(lista\_opciones\_valores) == 1:

tipo\_caja = random.choices(["rapida", "normal"], weights = [1, 0])

else:

tipo\_caja = [None]

tipe = tipo\_caja[0]

return tipo\_caja[0]

def proximo\_evento(self):

minimo = p.INFINITO

for evento in self.dict\_tiempo\_proximo.keys():

if self.dict\_tiempo\_proximo[evento] <= minimo:

minimo = self.dict\_tiempo\_proximo[evento]

proxima\_accion = evento

return proxima\_accion, minimo

def verificar\_proximo\_evento(self):

lista = []

if self.estado == "abierto":

lista.append("llegada")

if self.cantidad\_clientes\_comprando > 0:

lista.append("compra")

if self.cantidad\_clientes\_caja > 0:

lista.append("atencion")

return lista

def actualizar\_dict\_tiempo\_anterior(self, evento, tiempo):

self.dict\_evento\_anterior["evento"] = evento

self.dict\_evento\_anterior["tiempo"] = tiempo

self.dict\_tiempo\_proximo = {"llegada": p.INFINITO, "compra": p.INFINITO, "atencion": p.INFINITO}

def ocurre\_evento(self, evento, tiempo\_evento, tiempo\_actual):

if evento == "llegada":

self.cantidad\_clientes\_comprando += 1

self.cantidad\_llegadas += 1

self.tiempos\_llegadas.append(tiempo\_evento)

elif evento == "compra":

self.cantidad\_clientes\_comprando -= 1

self.cantidad\_clientes\_caja += 1

self.tiempos\_fin\_compra.append(tiempo\_evento)

self.tipo\_caja\_actual = self.elegir\_tipo\_caja(evento)

id\_caja = self.cola\_caja\_mas\_corta()

self.dict\_cajas[self.tipo\_caja\_actual][id\_caja].cola\_caja += 1

self.dict\_cajas[self.tipo\_caja\_actual][id\_caja].tiempo\_llegada\_cola.put(tiempo\_actual)

elif evento == "atencion":

self.cantidad\_clientes\_caja -= 1

self.cantidad\_atenciones += 1

instancia\_caja = self.dict\_cajas[self.tipo\_caja\_actual][self.caja\_rapida]

instancia\_caja.cola\_caja -= 1

tiempo\_llegada = instancia\_caja.tiempo\_llegada\_cola.get()

espera = tiempo\_actual - tiempo\_evento - tiempo\_llegada

self.dict\_tiempo\_espera\_caja[self.tipo\_caja\_actual][instancia\_caja.id\_caja].append(espera)

self.agregar\_tiempo\_ocupacion(tiempo\_evento)

def cola\_caja\_mas\_corta(self):

dict\_caja\_cola = {}

for tipo\_caja in self.dict\_cajas.keys():

for caja in self.dict\_cajas[tipo\_caja].values():

dict\_caja\_cola[caja.cola\_caja] = (caja.id\_caja, tipo\_caja)

menor\_cola = min(dict\_caja\_cola.keys())

id\_caja = dict\_caja\_cola[menor\_cola][0]

tipo = dict\_caja\_cola[menor\_cola][1]

self.tipo\_caja\_actual = tipo

minimo\_largo = self.dict\_cajas[self.tipo\_caja\_actual][id\_caja].cola\_caja

lista\_eleccion = []

for caja in self.dict\_cajas[self.tipo\_caja\_actual].values():

if caja.cola\_caja == minimo\_largo:

lista\_eleccion.append(caja)

eleccion = random.choice(lista\_eleccion)

id\_eleccion = eleccion.id\_caja

id\_caja = id\_eleccion

return id\_caja

def agregar\_tiempo\_ocupacion(self, tiempo):

for tipo\_caja in self.dict\_ocupacion\_caja.keys():

for llave in self.dict\_cajas[tipo\_caja].keys():

if self.dict\_cajas[tipo\_caja][llave].cola\_caja > 0:

self.dict\_ocupacion\_caja[tipo\_caja][llave] += tiempo

Caja

from queue import Queue

class Caja:

def \_\_init\_\_(self, id, tipo):

self.id\_caja = id

self.cola\_caja = 0

self.tipo = tipo

self.estado = "vacia"

self.tiempo\_llegada\_cola = Queue()

Resultados

from guardar\_datos import cargar\_datos

import parametros as p

def valor\_replica(ruta, rep):

replica = str(rep - 1)

dict\_ = cargar\_datos(ruta)

for caja in dict\_[replica].keys():

valor = dict\_[replica][caja]

redondeo = round(valor, 2)

def valor\_caja(ruta):

dict\_ = cargar\_datos(ruta)

for caja in dict\_.keys():

valor = dict\_[caja]

redondeo = round(valor, 2)

def promedios\_error(ruta):

dict\_ = cargar\_datos(ruta)

suma\_normal = 0

suma\_rapida = 0

for caja in dict\_.keys():

error = dict\_[caja]

if 0 <= int(caja) < p.CANTIDAD\_CAJAS\_NORMALES:

suma\_normal += error

else:

suma\_rapida += error

error\_normal = round(suma\_normal / p.CANTIDAD\_CAJAS\_NORMALES, 2)

error\_rapida = round(suma\_rapida / p.CANTIDAD\_CAJAS\_RAPIDAS, 2)

def promedios\_intervalos(ruta):

dict\_ = cargar\_datos(ruta)

suma\_normal\_min = 0

suma\_rapida\_min = 0

suma\_normal\_max = 0

suma\_rapida\_max = 0

for caja in dict\_.keys():

intervalo = dict\_[caja]

minimo = intervalo[0]

maximo = intervalo[1]

if 0 <= int(caja) < p.CANTIDAD\_CAJAS\_NORMALES:

suma\_normal\_min += minimo

suma\_normal\_max += maximo

else:

suma\_rapida\_min += minimo

suma\_rapida\_max += maximo

normal\_min = round(suma\_normal\_min / p.CANTIDAD\_CAJAS\_NORMALES, 2)

normal\_max = round(suma\_normal\_max / p.CANTIDAD\_CAJAS\_NORMALES, 2)

rapida\_min = round(suma\_rapida\_min / p.CANTIDAD\_CAJAS\_RAPIDAS, 2)

rapida\_max = round(suma\_rapida\_max / p.CANTIDAD\_CAJAS\_RAPIDAS, 2)

for i in range(11, 15):

promedios\_error(p.LISTA\_PATH[i])

for j in range(7, 11):

promedios\_intervalos(p.LISTA\_PATH[j])

Parametros

import json  
def abrir\_json(nombre\_archivo):

try:

with open(nombre\_archivo, 'r') as archivo:

datos = json.load(archivo)

return datos

except FileNotFoundError:

print(f"El archivo '{nombre\_archivo}' no fue encontrado.")

return None

except json.JSONDecodeError:

print(f"Error al decodificar el JSON en '{nombre\_archivo}'.")

return None

DICCIONARIO\_DATOS = abrir\_json("jumbo/parametros.json")

DICCIONARIO\_DATOS\_LLEGADAS = abrir\_json("jumbo/parametros\_llegada.json")

TASA\_COMPRA = DICCIONARIO\_DATOS["TIEMPO\_COMPRANDO"]

TASA\_ATENCION\_NORMAL = DICCIONARIO\_DATOS["TIEMPO\_CAJA\_NORMAL"]

TASA\_ATENCION\_RAPIDA = DICCIONARIO\_DATOS["TIEMPO\_CAJA\_RAPIDA"]

TASA\_ATENCION = {"normal": TASA\_ATENCION\_NORMAL, "rapida": TASA\_ATENCION\_RAPIDA}

CANTIDAD\_CAJAS\_NORMALES = 33 # NUEVO PARAMETRO

CANTIDAD\_CAJAS\_RAPIDAS = 12 # NUEVO PARAMETRO

PROBABILIDAD\_ELEGIR\_CAJA\_RAPIDA = (CANTIDAD\_CAJAS\_RAPIDAS) / (CANTIDAD\_CAJAS\_NORMALES + CANTIDAD\_CAJAS\_RAPIDAS)

PROBABILIDAD\_ELEGIR\_CAJA\_NORMAL = 1 - PROBABILIDAD\_ELEGIR\_CAJA\_RAPIDA

MINUTOS\_POR\_HORA = 60

JORNADA = 13

REPLICAS = 30

INFINITO = 9999999999999999999

LISTA\_PATH = ["datos\espera\_replicas.json",

"datos\ocupacion\_replicas.json",

"datos\dict\_tiempo\_de\_ejecucion.json",

"datos\dict\_tiempo\_promedio\_espera\_cola.json",

"datos\dict\_porcentaje\_ocupacion\_cajas.json",

"datos\sd\_espera.json",

"datos\sd\_ocupacion.json",

"datos\int\_90\_espera.json",

"datos\int\_90\_ocupacion.json",

"datos\int\_95\_espera.json",

"datos\int\_95\_ocupacion.json",

"datos\error\_90\_espera.json",

"datos\error\_95\_espera.json",

"datos\error\_90\_ocupacion.json",

"datos\error\_95\_ocupacion.json"]

## No homogeneo

import parametros as p

def tasa\_no\_homogeneo(minutos):

dict\_llegada = p.DICCIONARIO\_DATOS\_LLEGADAS

hora = int(minutos / p.MINUTOS\_POR\_HORA)

tasa\_llegada = dict\_llegada[str(hora)]

return tasa\_llegada

## Intervalos

from guardar\_datos import cargar\_datos

import parametros as p

import math

class Intervalo:

def \_\_init\_\_(self):

self.dict\_espera = cargar\_datos(p.LISTA\_PATH[3])

self.dict\_ocupacion = cargar\_datos(p.LISTA\_PATH[4])

self.sd\_espera = cargar\_datos(p.LISTA\_PATH[5])

self.sd\_ocupacion = cargar\_datos(p.LISTA\_PATH[6])

self.int\_espera\_90 = {}

self.int\_ocupacion\_90 = {}

self.int\_espera\_95 = {}

self.int\_ocupacion\_95 = {}

self.error\_90\_espera = {}

self.error\_90\_ocupacion = {}

self.error\_95\_espera = {}

self.error\_95\_ocupacion = {}

def definir\_valor\_t(self, confianza):

if confianza == 90:

return 1.697

elif confianza == 95:

return 2.042

def calcular\_intervalo\_espera(self, confianza):

for caja in self.dict\_espera.keys():

promedio = self.dict\_espera[caja]

sd = self.sd\_espera[caja]

sqrt\_n = math.sqrt(p.REPLICAS)

valor\_t = self.definir\_valor\_t(confianza)

min = promedio - (valor\_t \* (sd / sqrt\_n))

max = promedio + (valor\_t \* (sd / sqrt\_n))

intervalo = [min, max]

self.calcular\_error\_espera(min, max, caja, confianza)

if confianza == 90:

self.int\_espera\_90[caja] = intervalo

elif confianza == 95:

self.int\_espera\_95[caja] = intervalo

if confianza == 90:

return self.int\_espera\_90

elif confianza == 95:

return self.int\_espera\_95

def calcular\_intervalo\_ocupacion(self, confianza):

for caja in self.dict\_ocupacion.keys():

promedio = self.dict\_ocupacion[caja]

sd = self.sd\_ocupacion[caja]

sqrt\_n = math.sqrt(p.REPLICAS)

valor\_t = self.definir\_valor\_t(confianza)

min = promedio - (valor\_t \* (sd / sqrt\_n))

max = promedio + (valor\_t \* (sd / sqrt\_n))

intervalo = [min, max]

self.calcular\_error\_ocupacion(min, max, caja, confianza)

if confianza == 90:

self.int\_ocupacion\_90[caja] = intervalo

elif confianza == 95:

self.int\_ocupacion\_95[caja] = intervalo

if confianza == 90:

return self.int\_ocupacion\_90

elif confianza == 95:

return self.int\_ocupacion\_95

def calcular\_error\_espera(self, min, max, llave, confianza):

promedio = self.dict\_espera[llave]

diferencia = max - min

mitad = (max - min) / 2

error = (mitad) \* 100 / promedio

if confianza == 90:

self.error\_90\_espera[llave] = error

elif confianza == 95:

self.error\_95\_espera[llave] = error

def calcular\_error\_ocupacion(self, min, max, llave, confianza):

promedio = self.dict\_ocupacion[llave]

mitad = (max - min) / 2

error = (mitad) \* 100 / promedio

if confianza == 90:

self.error\_90\_ocupacion[llave] = error

elif confianza == 95:

self.error\_95\_ocupacion[llave] = error

Guardar datos

import parametros as ps

import json

def crear\_archivo():

for path in ps.LISTA\_PATH:

with open(path, "w") as archivo:

datos\_vacios = {}

json.dump(datos\_vacios, archivo, indent=4)

def cargar\_datos(path):

with open(path, "r") as archivo:

diccionario = json.load(archivo)

return diccionario

def guardar\_resultados(path, texto):

contenido = cargar\_datos(path)

contenido.update(texto)

with open(path, "w") as archivo:

json.dump(contenido, archivo, indent=4)

Excel

from guardar\_datos import cargar\_datos

import parametros as p

import pandas as pd

dict\_espera = cargar\_datos(p.LISTA\_PATH[3])

dict\_ocupacion = cargar\_datos(p.LISTA\_PATH[4])

dict\_sd\_espera = cargar\_datos(p.LISTA\_PATH[5])

dict\_sd\_ocupacion = cargar\_datos(p.LISTA\_PATH[6])

dict\_int\_90\_espera = cargar\_datos(p.LISTA\_PATH[7])

dict\_int\_90\_ocupacion = cargar\_datos(p.LISTA\_PATH[8])

dict\_int\_95\_espera = cargar\_datos(p.LISTA\_PATH[9])

dict\_int\_95\_ocupacion = cargar\_datos(p.LISTA\_PATH[10])

dict\_err\_90\_espera = cargar\_datos(p.LISTA\_PATH[11])

dict\_err\_90\_ocupacion = cargar\_datos(p.LISTA\_PATH[12])

dict\_err\_95\_espera = cargar\_datos(p.LISTA\_PATH[13])

dict\_err\_95\_ocupacion = cargar\_datos(p.LISTA\_PATH[14])

dict\_espera\_replicas = cargar\_datos(p.LISTA\_PATH[0])

dict\_ocupacion\_replicas = cargar\_datos(p.LISTA\_PATH[1])

df\_espera = pd.DataFrame(list(dict\_espera.items()), columns=["Caja", "Tiempo de Espera [min] promedio"])

df\_espera.to\_excel("excel/tiempos\_espera\_promedio.xlsx", index=False)

df\_ocupacion = pd.DataFrame(list(dict\_ocupacion.items()), columns=["Caja", "Porcentaje de Ocupación [%] promedio"])

df\_ocupacion.to\_excel("excel/porcentaje\_ocupacion\_promedio.xlsx", index=False)

df\_sd\_espera = pd.DataFrame(list(dict\_sd\_espera.items()), columns=["Caja", "Dev. Estándar Espera"])

df\_sd\_espera.to\_excel("excel/sd\_espera.xlsx", index=False)

df\_sd\_ocupacion = pd.DataFrame(list(dict\_sd\_ocupacion.items()), columns=["Caja", "Dev. Estándar Ocupación"])

df\_sd\_ocupacion.to\_excel("excel/sd\_ocupacion.xlsx", index=False)

df\_int\_90\_espera = pd.DataFrame(dict\_int\_90\_espera.items(), columns=['Caja', 'Intervalo Espera [min] al 90%'])

df\_int\_90\_espera.to\_excel('excel/int\_90\_espera.xlsx', index=False)

df\_int\_95\_espera = pd.DataFrame(dict\_int\_95\_espera.items(), columns=['Caja', 'Intervalo Espera [min] al 95%'])

df\_int\_95\_espera.to\_excel('excel/int\_95\_espera.xlsx', index=False)

df\_int\_90\_ocupacion = pd.DataFrame(dict\_int\_90\_ocupacion.items(), columns=['Caja', 'Intervalo Ocupación [%] al 90%'])

df\_int\_90\_ocupacion.to\_excel('excel/int\_90\_ocupacion.xlsx', index=False)

df\_int\_95\_ocupacion = pd.DataFrame(dict\_int\_95\_ocupacion.items(), columns=['Caja', 'Intervalo Ocupación [%] al 95%'])

df\_int\_95\_ocupacion.to\_excel('excel/int\_95\_ocupacion.xlsx', index=False)

df\_err\_90\_esperado = pd.DataFrame(dict\_err\_90\_espera.items(), columns=['Caja', 'Error Espera al 90%'])

df\_err\_90\_esperado.to\_excel('excel/err\_90\_espera.xlsx', index=False)

df\_err\_90\_ocupacion = pd.DataFrame(dict\_err\_90\_ocupacion.items(), columns=['Caja', 'Error Ocupación al 90%'])

df\_err\_90\_ocupacion.to\_excel('excel/err\_90\_ocupacion.xlsx', index=False)

df\_err\_95\_esperado = pd.DataFrame(dict\_err\_95\_espera.items(), columns=['Caja', 'Error Espera al 95%'])

df\_err\_95\_esperado.to\_excel('excel/err\_95\_espera.xlsx', index=False)

df\_err\_95\_ocupacion = pd.DataFrame(dict\_err\_95\_ocupacion.items(), columns=['Caja', 'Error Ocupación al 95%'])

df\_err\_95\_ocupacion.to\_excel('excel/err\_95\_ocupacion.xlsx', index=False)

replicas = []

cajas = []

valores = []

for replica, data in dict\_espera\_replicas.items():

for caja, valor in data.items():

replicas.append(replica)

cajas.append(caja)

valores.append(valor)

df\_espera\_replicas = pd.DataFrame({

'Réplica': replicas,

'Caja': cajas,

'Valor [min]': valores})

df\_espera\_replicas.to\_excel("excel/espera\_replicas.xlsx", index=False)

replicas = []

cajas = []

valores = []

for replica, data in dict\_ocupacion\_replicas.items():

for caja, valor in data.items():

replicas.append(replica)

cajas.append(caja)

valores.append(valor)

df\_ocupacion\_replicas = pd.DataFrame({

'Réplica': replicas,

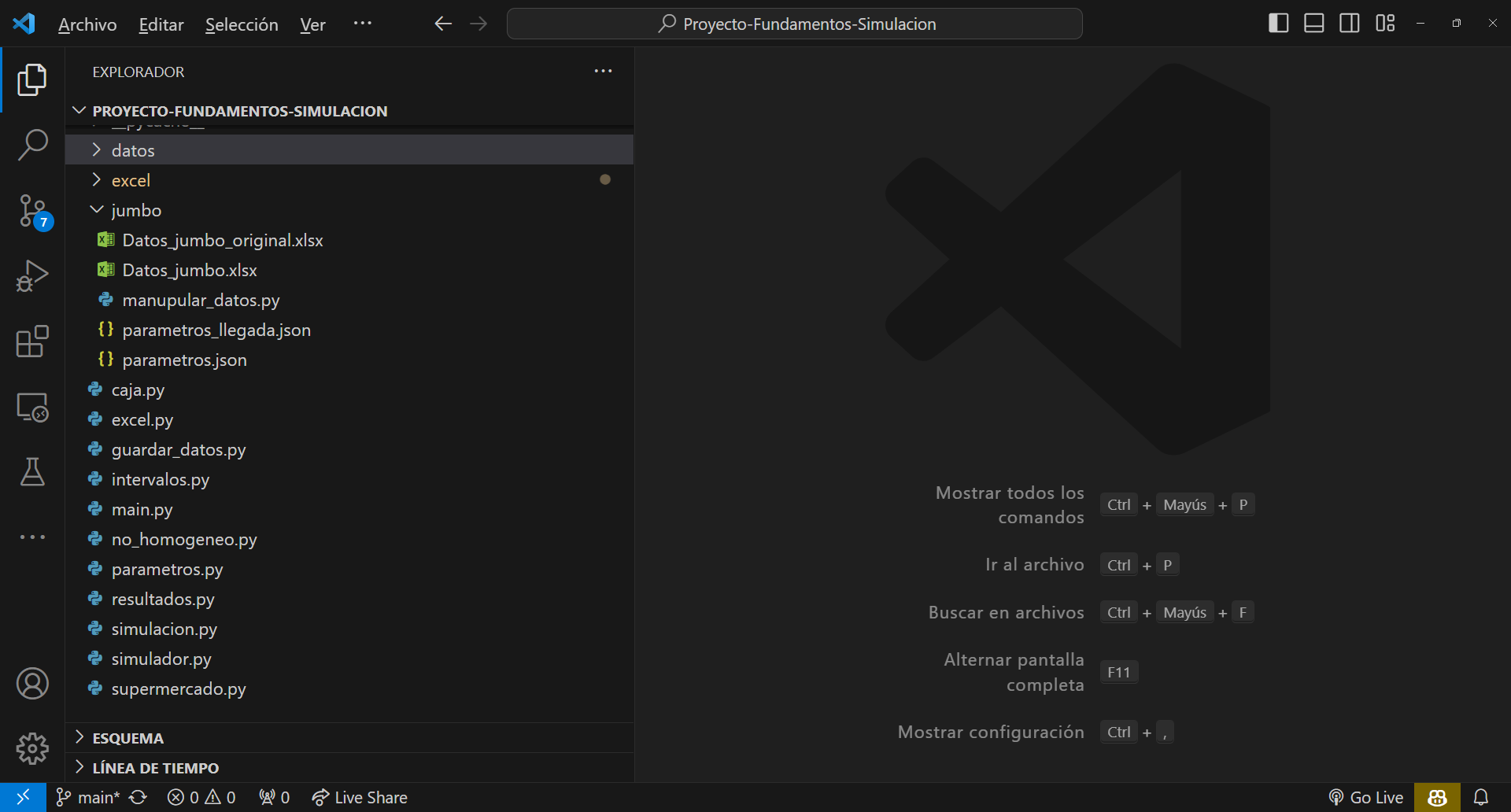
'Caja': cajas,

'Valor [%]': valores})

df\_ocupacion\_replicas.to\_excel("excel/ocupacion\_replicas.xlsx", index=False)

## Python

Repositorio: <https://github.com/TriniAguilera/Proyecto-Fundamentos-Simulacion.git>



# Anexos

|  |  |
| --- | --- |
| TIEMPOS DE COMPRA CAPTURADOS EN EL SUPERMERCADO | |
| TIEMPO DE COMPRA | TIEMPO DE COMPRA |
| 0:11:30 | 0:08:59 |
| 0:20:54 | 0:40:14 |
| 0:07:40 | 0:34:02 |
| 0:00:28 | 0:45:00 |
| 0:18:28 | 0:20:00 |
| 0:31:40 | 0:24:00 |
| 0:28:42 | 0:06:30 |
| 0:47:41 |  |

Tabla 10: Tiempos de compra del supermercado

|  |  |  |
| --- | --- | --- |
| TIEMPOS DE ATENCION CAJAS NORMAL CAPTURADOS EN EL SUPERMERCADO | | |
| TIEMPO CAJA ATENCION CAJA NORMAL | TIEMPO CAJA ATENCION CAJA NORMAL | TIEMPO CAJA ATENCION CAJA NORMAL |
| 0:00:35 | 0:02:05 | 0:00:52 |
| 0:00:54 | 0:02:12 | 0:02:10 |
| 0:00:44 | 0:01:23 | 0:04:42 |
| 0:00:52 | 0:00:51 | 0:01:05 |
| 0:02:05 | 0:02:03 | 0:01:26 |
| 0:00:46 | 0:06:15 | 0:00:59 |
| 0:01:19 | 0:02:14 | 0:01:12 |
| 0:00:45 | 0:02:03 | 0:06:36 |
| 0:01:12 | 0:00:52 | 0:04:07 |
| 0:02:29 | 0:00:38 | 0:01:30 |
| 0:06:06 | 0:00:20 | 0:01:40 |
| 0:01:50 | 0:05:24 | 0:00:38 |
| 0:01:33 | 0:01:11 | 0:01:27 |
| 0:09:44 | 0:00:58 | 0:02:10 |
| 0:01:55 | 0:01:35 | 0:02:49 |
| 0:02:07 | 0:01:21 | 0:01:32 |
| 0:00:43 |  |  |

Tabla 11: Tiempos de atención de caja normal

|  |  |  |
| --- | --- | --- |
| TIEMPOS DE ATENCION CAJA RAPIDA CAPTURADOS EN EL SUPERMERCADO | | |
| TIEMPO ATENCIÓN CAJA RÁPIDA | TIEMPO ATENCIÓN CAJA RÁPIDA | TIEMPO ATENCIÓN CAJA RÁPIDA |
| 0:01:30 | 0:01:24 | 0:01:13 |
| 0:02:23 | 0:01:10 | 0:00:52 |
| 0:02:51 | 0:00:34 | 0:01:56 |
| 0:01:33 | 0:02:11 | 0:00:57 |
| 0:02:30 | 0:02:24 | 0:00:30 |
| 0:02:50 | 0:00:54 | 0:02:11 |
| 0:02:33 | 0:00:29 | 0:01:16 |
| 0:02:31 | 0:02:13 | 0:01:29 |
| 0:00:53 | 0:02:02 | 0:02:12 |
| 0:00:42 | 0:01:20 | 0:01:18 |
| 0:03:37 | 0:00:47 | 0:01:57 |
| 0:01:42 | 0:01:13 | 0:02:02 |
| 0:01:02 | 0:02:14 | 0:02:52 |
| 0:01:05 | 0:00:39 | 0:03:14 |
| 0:05:04 | 0:01:00 | 0:01:13 |
| 0:01:22 | 0:01:15 | 0:02:16 |
| 0:01:03 | 0:01:07 | 0:01:53 |
| 0:02:12 | 0:01:01 | 0:02:20 |
| 0:02:10 | 0:02:50 | 0:01:01 |
| 0:00:43 | 0:00:41 | 0:01:07 |
| 0:01:23 | 0:00:41 | 0:01:25 |
| 0:02:15 | 0:01:08 | 0:01:15 |
| 0:06:26 | 0:00:15 | 0:05:52 |
| 0:00:38 | 0:01:25 | ... |

Tabla 12: Tiempos de atención de caja rápida