



[191013] Diseño de un método de asignación de ubicaciones que minimice el *lead time* de la operación en el centro de distribución Tittán Totto

Sofía Catalina Rodríguez Giraldo ^{1 a,c}, Juan José Callejas Gamba ^{2 a,c}, Daniela Rosado Cuervo ^{3 a,c}, Luis Alfonso Montoya Castillo ^{4 a,c}
Mohamed Rabie Nait-Abdallah ^{b,c}

^aEstudiante de Ingeniería Industrial

^bProfesor, Director del Proyecto de Grado, Departamento de Ingeniería Industrial

^cPontificia Universidad Javeriana, Bogotá, Colombia

Resumen de diseño en Ingeniería

One of the most expensive activities that make up *lead time* in warehouses is order picking. Storage allocation decisions (*slotting*) have a direct impact on the efficiency of order picking processes since the routes to collect items are affected by the location of the SKUs in the picking area. Therefore, this document presents the methodology with which a *slotting* application was developed. This application reduced the theoretical *lead time* of the operation by 29% for the picking area of the Tittan Totto warehouse. This project addressed three key components of *slotting*: the ordering strategy of SKUs, using heuristic methods such as the correlation strategy, and metaheuristics such as the taboo search and the genetic algorithm, the calculation of the number of *slots* that must be assigned to each SKU, using a mathematical model, and the maintenance policy to update the *slotting* according to the demand requirements. The solution given by the developed methods was evaluated and compared to the Totto solution, using real and hypothetical data from three months of operation. Finally, the results obtained were discussed.

Key words: Lead time, order preparation time, slotting, correlations, taboo search, genetic algorithm, dynamic slots, maintenance.

1. Justificación y planteamiento del problema

El *lead time* se entiende como el tiempo en el que se desarrollan los procesos que componen la cadena de suministro de una empresa. Una reducción de este tiempo podría resultar en la disminución de costos y en el aumento del valor agregado que se ofrece a los consumidores [1].

En los centros de distribución de las organizaciones se encuentra una de las actividades más laboriosas y costosas que componen el *lead time*, el alistamiento de pedidos o *picking*. El costo de este proceso puede alcanzar hasta un 55% de los costos totales de un centro de distribución (CEDI) [2] y, de éste, casi la mitad de los recursos son consumidos por el desplazamiento que se efectúa entre cada ubicación de producto [3]. Este problema se puede solucionar mediante la ubicación estratégica de los productos o *slotting*, debido a que reduce la distancia total recorrida por los colaboradores [4], el número de reabastecimientos y, en consecuencia, logra la disminución del *lead time* de la operación.

Descripción de la empresa

Totto es una multinacional colombiana líder en Latinoamérica con más de 30 años de experiencia en el sector de la moda. Esta empresa ofrece variedad y tecnología en maletines, morrales, mochilas, bolsos, ropa y accesorios. Su misión es convertirse en una organización líder en el desarrollo, producción y comercialización de productos de excelente calidad, dentro de un mercado global. Por lo anterior, para Totto, es vital la evaluación y mejora continua de sus procesos.

Descripción del Centro de Distribución

Su principal centro de distribución a nivel nacional es Tittán Totto, ubicado en Mosquera, Cundinamarca, el cual está organizado en 6 zonas diferentes (ver ilustración 1). Éstas son la zona de *picking* o Torre (A), la zona de almacenamiento (B), la zona de recibo y despacho (C), la zona de empaque y etiquetado (D), la zona de modificaciones (E) y la zona de *e-commerce* (F). Como se verá más adelante en el documento, las zonas diferentemente involucradas en el presente proyecto son la zona de *picking* y de almacenamiento.

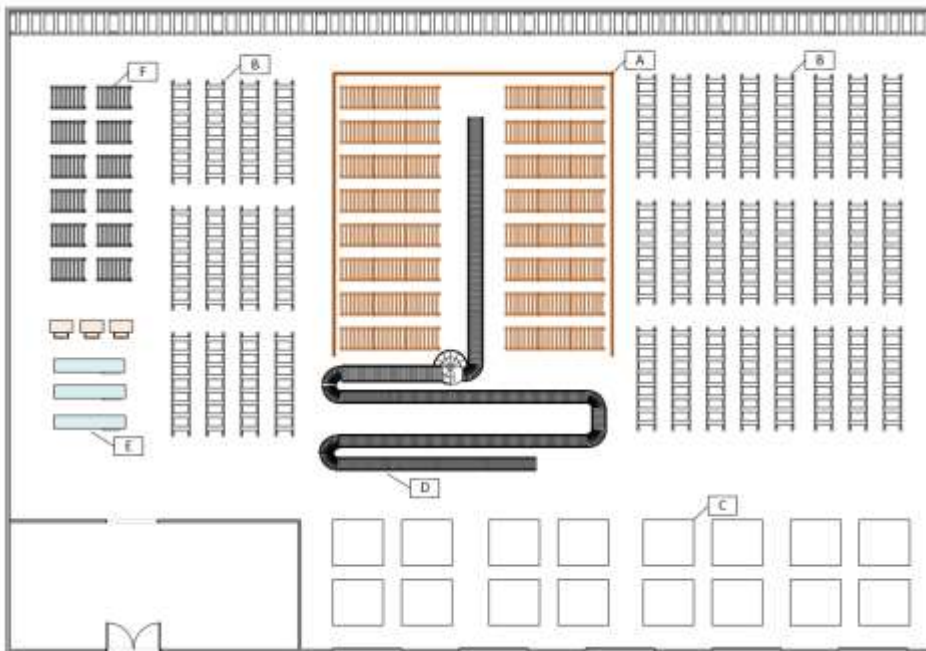


Ilustración 1. Plano actual del Centro de Distribución Tittán. Elaboración propia

Por un lado, en la zona de almacenamiento se aloja el inventario necesario para abastecer a la Torre además de las referencias de mayor volumen, baja rotación, discontinuadas y sobrantes de colecciones pasadas (saldos). Por otro lado, en la Torre se albergan los productos de mayor rotación y se lleva a cabo el proceso de alistamiento de pedidos. Esta estructura cuenta con cuatro pisos, cada uno dedicado al almacenamiento de conjuntos de productos cuyas características son similares (unidades de negocio): morrales, accesorios y ropa.

En cada piso, los productos son almacenados en *racks*, cada uno con determinada cantidad de *slots* (ubicaciones). Se manejan dos tipos de *slots*: fijos y dinámicos. Los *slots* fijos son ubicaciones a las que se les asigna un SKU permanentemente durante un periodo de tiempo específico. Los *slots* dinámicos son espacios que se dejan sin asignación fija en la Torre. Dada la posibilidad de alojar cualquier SKU que lo requiera, estas ubicaciones ayudan a soportar la incertidumbre de la demanda durante la temporada.

Lead time de la operación

El *lead time* de la operación se entiende como la latencia entre el inicio y la ejecución de un proceso [1]. Debido a que Totto maneja una estrategia *make to stock*, el *lead time* de la operación en el CEDI comprende el tiempo transcurrido entre la llegada de un pedido al centro de distribución y su preparación para enviar en los camiones. Los procesos que hacen parte de este tiempo y su proporción de incidencia en el *lead time* se pueden observar en la ilustración 2 (*ver ilustración 2*).

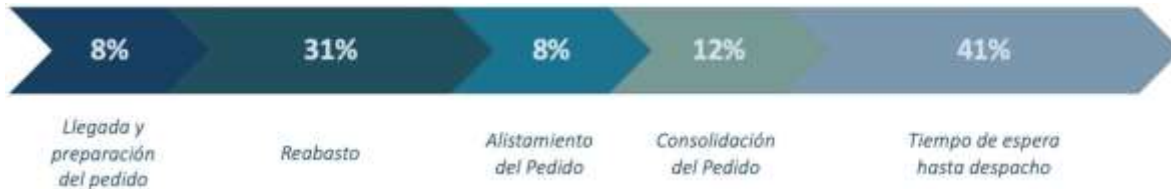


Ilustración 2. Composición del *lead time* de la operación. Elaboración propia

A partir de la composición del *lead time*, el reabastecimiento y el alistamiento de pedidos representan un 39% del tiempo total en el CEDI. Este proyecto se enfocó en disminuir estos dos tiempos.

Tiempo de reabasto

El tiempo de reabasto es el tiempo empleado en llevar a cabo una ola de reabasto. Esta consiste en el aprovisionamiento de la Torre con los Stock Keeping Units (SKUs), desde la zona de almacenamiento del CEDI, suficientes para suplir la necesidad de una ola de *picking*. Una ola de *picking* se compone de un conjunto de pedidos que van a ser recopilados de manera simultánea. Por consiguiente, antes de liberar una ola de reabasto, se evalúa si la cantidad de producto disponible en la Torre sufre en su totalidad la cantidad demandada en la ola de *picking*.

Tiempo de alistamiento de pedidos

El tiempo de alistamiento de pedidos se compone de la duración del tiempo de *pick* y el tiempo de viaje. El tiempo de *pick* es el tiempo empleado en extraer las unidades demandadas de sus ubicaciones, este tiempo es un valor estándar por unidad de producto. Mientras que el tiempo de viaje es la duración del desplazamiento entre ubicaciones durante el alistamiento de pedidos.

Para llevar a cabo el alistamiento, cada pedido perteneciente a la ola de *picking* es dividido en unidades de trabajo (UT). Una UT es un subconjunto de la orden que contiene aquellos productos que pertenecen a la misma unidad de negocio y tiene una capacidad máxima de 200 unidades. Una vez creadas las UTs del pedido, cada una es asignada a un colaborador. Siguiendo las políticas de la empresa, en el viaje de recolección, los colaboradores deben seguir una ruta en forma de U para recorrer los pisos de la Torre y en forma de N para recorrer los pasillos.

Slotting

El *slotting* es el proceso de asignar SKUs a ubicaciones dentro de una bodega de almacenamiento [5]. Este tiene como objetivo reducir el tiempo de viaje y minimizar la cantidad de veces que un SKU es reabastecido, incrementando la utilización de los recursos físicos y humanos [6].

En el CEDI Tittán Totto, se realiza un proceso de *slotting* en la Torre al inicio de cada temporada de productos, las cuales ocurren cada tres meses. Por lo tanto, el *slotting* determina las ubicaciones de los SKUs que cumplirán con las necesidades de la demanda para cada temporada del año.

Las actividades que se realizan para llevar a cabo este proceso son:

1. Determinar el ordenamiento de los SKUs dentro de la Torre.
2. Determinar la cantidad de *slots* fijos para cada SKU y el porcentaje de *slots* dinámicos que tendrá la Torre.

Actualmente, la forma en que Tutto toma las decisiones mencionadas anteriormente es la siguiente:

1. Para el ordenamiento, la compañía se basa en una política ABC, en la cual los SKUs son ordenados en función de la información de la demanda de los meses pasados, de forma que la categoría A reúne los SKUs más demandados y, por lo tanto, son los primeros en el ordenamiento, mientras que la C reúne los SKUs de menor demanda y, por lo tanto, se encuentran al final del ordenamiento.
2. De acuerdo con los datos históricos, se hace una aproximación empírica de la cantidad de *slots* fijos que deben ser asignados a cada SKU. Además, el 30% de las ubicaciones de la Torre deben ser *slots* dinámicos.

Mantenimiento del slotting

Las políticas de mantenimiento buscan ajustar periódicamente el *slotting* a las necesidades que genere la demanda a lo largo del horizonte de tiempo, de manera que se asegure un adecuado uso del espacio.

Actualmente, para asegurar un uso adecuado del espacio en la Torre, se lleva a cabo un mantenimiento de *slots* dinámicos. Este proceso consiste en hacer la liberación del mayor número posible de *slots* dinámicos mediante el traslado de las unidades que se encuentran en un dinámico a un fijo previamente asignado.

Adicionalmente, la empresa busca desarrollar una política de mantenimiento o *reslotting* en un periodo de menor duración con respecto al del *slotting*, en el que se ajuste el *slotting* inicial de acuerdo con la variabilidad de la demanda.

Restricciones del slotting

El proceso de *slotting* cuenta con algunas restricciones, debido a las características de la empresa y del CEDI. Estas se presentan a continuación:

1. *Disponibilidad de espacio*: actualmente, la empresa cuenta con aproximadamente 23.000 *slots* dentro de la Torre frente a los 17.000 SKUs que se manejan.
2. A cada *slot* se le asigna únicamente un SKU, es decir, los *slots* son mono referencia.
3. Cada *slot* tiene una capacidad específica para cada SKU.
4. *Demanda*: no se admiten faltantes.
5. *Ergonomía*: los productos de gran tamaño y/o peso deben ser ubicados en los niveles uno y dos de los racks. Además, se busca que los productos de baja rotación se ubiquen en el tercer y cuarto piso de la Torre. Esto se hace con el objetivo minimizar los accidentes y/o riesgos a los que puedan estar expuestos los colaboradores.

Enfoque del proyecto

Este proyecto se enfocó en minimizar los tiempos de alistamiento de pedidos y reabasto mediante el desarrollo de una metodología de asignación de ubicaciones (*slotting*) dentro la Torre del CEDI Tittán Tutto. El *slotting* se abordó como un problema de optimización de operaciones que considera las siguientes decisiones:

- Estrategia de ordenamiento de los SKUs en la Torre
- Cantidad de *slots* fijos que se debe asignar a cada SKU y porcentaje de *slots* dinámicos en la Torre

- Política de mantenimiento del *slotting*

Teniendo en cuenta lo anterior, este proyecto tiene como propósito responder la siguiente pregunta: *¿Cómo diseñar un método de asignación de ubicaciones que minimice el lead time de la operación en el centro de distribución Tittán Tutto?*

2. Antecedentes

Introducción

En esta sección se presenta una revisión de los métodos empleados por diferentes autores para minimizar el *lead time* de la operación de un centro de distribución mediante la disminución del tiempo de alistamiento de pedidos. Según [7], las decisiones de asignación de almacenamiento (*slotting*) tienen un impacto directo en la eficiencia de los procesos de preparación de pedidos, ya que las rutas de preparación se ven afectadas por la ubicación relativa de los SKU en un pedido.

Método óptimo

En [8], los autores presentaron un modelo matemático para minimizar el tiempo total de viaje durante el proceso de preparación de pedidos para los SKUs en un centro de distribución con pasillos rectangulares y cruces. La única variable de este modelo decide si el SKU i es ubicado en el slot j . La particularidad de este modelo es que toma como costo de la función objetivo la cantidad de veces que se visita una ubicación multiplicado por la distancia recorrida desde las ubicaciones hasta el punto de referencia. Lo anterior se debe a que no sería preciso calcular la distancia total recorrida en términos de la cantidad de producto demandada, sino a la frecuencia con la que se visita una ubicación. Como resultado, en una situación simulada con tres réplicas de un experimento en los que se re-slotteaba el 100% de los slots, se obtuvo una reducción del tiempo de viaje de alrededor del 30% y, en consecuencia, la reducción del tiempo de alistamiento de pedido.

Métodos heurísticos clásicos

Se ha demostrado que el problema de asignación de ubicaciones de almacenamiento es un problema NP-Hard [9] incluso en escenarios a escala, por lo que se suele abarcar este problema mediante diferentes soluciones heurísticas. Según [7] [10], desde una perspectiva de control de planificación de operaciones se tienen 3 políticas de almacenamiento muy comunes: la aleatoria (randomized storage policies), la dedicada (dedicated storage policies) y la de clasificación (class-based storage policies).

Políticas de almacenamiento aleatorio y dedicado

Para [7], las políticas de almacenamiento aleatorio asignan SKUs entrantes a ubicaciones disponibles al azar, mientras que, según [11] en el almacenamiento dedicado, cada artículo tiene una ubicación de almacenamiento fija que se deja vacía si el artículo está agotado. La primera política, según [10], provoca una alta utilización del almacenamiento y una baja eficiencia en la preparación de pedidos. Por el contrario, según este mismo autor, la segunda estrategia resulta en que los recolectores se familiarizan con la ubicación de los artículos, por lo que la eficiencia de la recolección mejora al precio de la baja utilización del almacenamiento.

Según [10], en la estrategia dedicada, se asignan las ubicaciones de acuerdo con las propiedades individuales de los SKUs, como popularidad, rotación, volumen o COI. [12], las describe como se encuentra a continuación:

- *Popularidad*: número de veces que se visitan los espacios en los que se encuentra una clase de elemento específica

- *Rotación*: cantidad total de un artículo enviado en un período de tiempo fijo. Los productos con un alto valor de rotación (productos de movimiento rápido) se asignan en las ranuras más cercanas a la entrada/salida. El principio opuesto se aplica a los productos de movimiento lento.
- *Volumen*: la demanda esperada de un producto específico en un período de tiempo específico multiplicado por su volumen.
- *Índice de cubo por pedido - COI*: relación entre el volumen ocupado por el producto específico y su popularidad. Los productos con un COI bajo se encuentran más cerca de las entradas/salidas. La estrategia opuesta se aplica a los productos con un COI alto.

Heurística de Clasificación

Según [13], la estrategia de clasificación es ampliamente utilizada en la práctica porque es conveniente de implementar y de mantener. De acuerdo con [7], los enfoques de asignación de almacenamiento basados en la clase tienden a disminuir el tiempo del proceso de preparación de pedidos al colocar los SKUs con una alta tasa de rotación en una ubicación que es más fácilmente accesible. Más específicamente, este tipo de ordenamiento asigna SKUs en función de sus tasas de demanda, clasificación ABC y almacenan SKU en regiones reservadas para su clase respectiva. En algunos casos, las decisiones de asignación de almacenamiento pueden involucrar múltiples criterios.

Heurística por correlaciones

La primera vez que se formuló la estrategia de asignación correlacionada fue en 1989 por [9]. Los autores afirman que los SKU que demandan juntos frecuentemente deben asignarse cerca uno del otro en el almacén, de modo que se puedan recoger más SKUs en menos tiempo. [14] demuestra que, como resultado del uso de esta estrategia, se reduce el esfuerzo de recolección y el tiempo de alistamiento de los pedidos en aproximadamente un 3% a comparación de la política dedicada con criterio de rotación completa. Sin embargo, según este mismo autor, cuando los artículos se demandan independientemente, las ventajas de las estrategias basadas en correlación desaparecen.

Métodos metaheurísticos

Búsqueda local

En este campo, se han llevado a cabo diferentes aproximaciones. Así pues, en [12], los autores desarrollan un modelo matemático para solucionar este problema, pero recomiendan una heurística de búsqueda local iterada cuando el número de clases de productos es muy elevado. En este caso, la heurística empieza con una solución generada a través de la asociación de un peso aleatorio a cada clase de producto, a la cual se realiza una determinada cantidad de movimientos locales buscando encontrar una mejor solución en el vecindario de la solución actual. Posteriormente, se genera una perturbación en la búsqueda, con el objetivo de encontrar mejores soluciones. El proceso es repetido por un número de iteraciones o tiempo determinado. Finalmente, este estudio resalta beneficios en tiempo y costos utilizando esta metodología.

Algoritmo Genético

Por otro lado, según [15], una de las metaheurísticas más utilizadas en la asignación de ubicaciones de SKUs son los algoritmos genéticos. El objetivo del método de solución de estos autores es disminuir la distancia recorrida para un conjunto de pedidos simulado. En primer lugar, se genera una solución inicial a partir de una asignación de ubicaciones basada en clases. Luego, se elige un patrón de picking preestablecido mediante el desarrollo de heurísticas, con el cual se calcula la distancia recorrida para cada posible solución. Entonces, se diseña e implementa el algoritmo genético en el que los alelos de los cromosomas representan los SKUs mientras

que los genes representan las ubicaciones dentro del almacén. Los resultados este algoritmo muestran una reducción del 20% de la distancia recorrida en relación con el método clásico de asignación basada en clases.

Ant Colony Optimization (ACO):

Desde otra perspectiva, Yingde y Smith [16] proponen realizar un slotting dinámico basado en las correlaciones de pedidos de los SKUs, con el objetivo de minimizar el tiempo máximo de duración del alistamiento de pedidos. Para determinar el slotting, los autores usaron la metaheurística ACO (Ant Colony Optimization) y para reducir el tiempo computacional, usaron una heurística que contiene la información de la correlación entre SKUs y el respectivo número de picks. Como resultado de esta propuesta, se obtuvo que al comparar la solución con la de la metodología COI (Cube per Order Index), el método ACO siempre proporciona una mejor función objetivo y que no es importante acomodar los productos con una fuerte correlación cerca el uno del otro, sino ubicarlos en la misma zona de picking.

Conclusión

Con base en la revisión de literatura, se observa un patrón metodológico al abordar la disminución del tiempo de alistamiento de pedidos mediante el slotting. Estos suelen comenzar con la determinación de una solución heurística inicial y, las más usadas son la aleatoria, la determinada, por clasificación y por correlaciones. Posteriormente, esta solución es mejorada a partir de la implementación metaheurísticas como la búsqueda local, algoritmos genéticos y los algoritmos ACO. Por tal motivo, se desarrolló un método de optimización de operaciones para la asignación de ubicaciones, el cual tuvo en cuenta los factores previamente señalados.

Por otro lado, en la revisión literaria realizada no se encontraron estudios que abarcaran a profundidad este problema considerando el uso de ubicaciones dinámicas ni la cantidad de slots que se deberían asignar a cada SKU. Estos son componentes primordiales para la implementación de un slotting en escenarios reales, de manera que, el método desarrollado incorporó estos temas.

3. Objetivos

Objetivo general

Desarrollar un aplicativo de asignación de ubicaciones de los SKUs en la Torre con el que disminuya el lead time de la operación en el centro de distribución Tittán Totto.

Objetivos específicos

- a. Determinar un método o métodos de optimización para la asignación de ubicaciones de los SKUs en Torre.
- b. Diseñar un aplicativo con el que se implemente el método o métodos de asignación de ubicaciones.
- c. Medir el impacto de la solución diseñada mediante la comparación del *lead time* de la operación actual y la propuesta.

4. Metodología

Para resolver este problema, se desarrollaron tres pasos enfocados en la disminución del tiempo de viaje (componente del tiempo de alistamiento de pedidos) y de reabasto para lograr la disminución del *lead time* de la operación.

El primer paso consistió en definir las estrategias de ordenamiento que se compararon respecto al tiempo de viaje. Además, se determinó el procedimiento para calcular este tiempo. La estrategia que más disminuyó el

tiempo de viaje se tomó como el ordenamiento para el método de asignación de ubicaciones (*slotting*) propuesto en este proyecto. Esto, debido a que el ordenamiento es la secuencia en la que son ubicados los SKUs y por lo tanto, es la base del recorrido para el alistamiento de pedidos.

El segundo paso consistió en desarrollar una estrategia de optimización para obtener la cantidad de *slots* fijos a asignar a cada SKU, de modo que se redujera el tiempo de reabasto, y un método para ajustar esta cantidad de *slots* a diferentes porcentajes de *slots* dinámicos. Además, el tercer paso consistió en determinar una política de mantenimiento que también redujera el tiempo de reabasto.

A continuación, se desarrolló un aplicativo de asignación de ubicaciones para consolidar diferentes escenarios de *slotting*, teniendo como entradas el mejor ordenamiento del paso uno y los escenarios de *slots* fijos y dinámicos generados por el paso dos. Posteriormente, se establecieron los panoramas de demanda para evaluar el impacto en el *lead time* de los escenarios consolidados junto con la política de mantenimiento propuesta en el paso tres.

Finalmente, se desarrolló un aplicativo que simula los procesos de la compañía y las aproximaciones respecto a la operación del CEDI, para calcular el *lead time* de cualquier escenario de *slotting* y la política de mantenimiento.

4.1. Estrategias de ordenamiento

De acuerdo con la revisión de literatura, el problema de establecer un ordenamiento para SKUs es un NP-Hard, por lo tanto, primero se definieron cuatro ordenamientos basados en diferentes heurísticas (aleatoria, correlaciones, multicriterio y ABC). Luego, se evaluaron los tiempos de viaje que generaron estos ordenamientos y se compararon. El ordenamiento con menor tiempo de viaje se utilizó como solución inicial para desarrollar dos metaheurísticas, con las cuales se pretendió minimizar este tiempo.

La comparación del tiempo de viaje generado por el ordenamiento de las heurísticas y metaheurísticas, se presentan en la sección de resultados.

4.1.1. Métodos heurísticos de ordenamiento

A continuación, se definen las heurísticas de ordenamiento a comparar. Para revisar el detalle del desarrollo de las primeras tres, ver el *Anexo 1*. La cuarta heurística es la que actualmente maneja Tutto.

Heurística de ordenamiento aleatoria

La primera heurística es un ordenamiento aleatorio, en el cual se da una posición aleatoria a cada SKU.

Heurística de ordenamiento basada en correlaciones

La segunda heurística es un ordenamiento por correlaciones. El objetivo de esta heurística es realizar un ordenamiento de forma que los SKUs que se suelen pedir con mayor frecuencia en los mismos pedidos estén más cerca entre sí. Asimismo, esta heurística ubica los SKUs con mayor demanda más cerca a la entrada de la zona de *picking* [14]. Para ello, se empleó el programa computacional MatLab, de acuerdo con el procedimiento realizado en [14]. Para revisar el aplicativo que se desarrolló, ver *Anexo 2*.

Heurística de ordenamiento multicriterio

La tercera heurística es un ordenamiento multicriterio. Debido a que las políticas de almacenamiento se enfocan en uno o varios criterios para ubicar los SKUs [7], en esta heurística se establecieron tres criterios con

los cuales se evaluó la demanda de las referencias para realizar el ordenamiento. Estos fueron los criterios empleados:

1. Cantidad de semanas en las que cada SKU fue demandado durante el periodo
2. Demanda total de cada SKU para el periodo
3. Cantidad de *slots* promedio a la semana para cada SKU durante el periodo

Heurística de ordenamiento por clasificación ABC (política manejada por Tutto)

La cuarta heurística es un ordenamiento por clasificación ABC. Este ordenamiento se explicó anteriormente en el apartado de *slotting*.

Procedimiento para el cálculo del tiempo de viaje en la comparación de ordenamientos

El tiempo de viaje generado por cada ordenamiento se calculó a partir de la distancia recorrida para recolectar los pedidos comprendidos entre mayo y julio, dividida entre la velocidad estándar de los colaboradores de la compañía.

Durante el periodo comprendido, no siempre se pidieron las mismas referencias y entraron nuevas que no estaban incluidas en el ordenamiento inicial. Por esto, se realizó un filtro en todos los pedidos para conservar únicamente los SKUs que hacían parte del ordenamiento inicial y, con base en estos, se calculó el tiempo de viaje. Además, para todos los ordenamientos se consideraron únicamente los SKUs presentes en el ordenamiento inicial de Tutto.

4.1.2. Métodos metaheurísticos

Con el objetivo de optimizar la función objetivo (tiempo de viaje) obtenida a partir del mejor ordenamiento heurístico, se consideraron tres estrategias de solución: búsqueda tabú, algoritmo genético y la combinación entre estas dos. Para estudiar el desempeño de estas, teniendo como solución inicial una heurística, se tomó como ejemplo la heurística de correlaciones.

En esta sección se describió individualmente cada estrategia y se analizó el rendimiento en la función objetivo de diferentes propuestas de aplicación para cada una. Además, se concluyó en cada estrategia qué propuesta de aplicación será la implementada para obtener una solución. [Por último, se realizó un análisis comparativo del desempeño de las metaheurísticas desarrolladas.](#)

Búsqueda Tabú

Según [17], la búsqueda Tabú es un procedimiento metaheurístico cuya característica distintiva es el uso de memoria adaptativa y de estrategias especiales de resolución de problemas. Según este autor, el propósito de esta metaheurística es guiar un procedimiento heurístico de búsqueda local, en este caso, el mejor ordenamiento heurístico, en la búsqueda de optimalidad global. A continuación, se describen los parámetros del algoritmo:

Input

La entrada a la metaheurística es la secuencia de ordenamiento de los SKUs de acuerdo con una de las cuatro heurísticas definidas en la sección anterior.

Propuestas para la aplicación de la búsqueda tabú

La solución inicial de la heurística, tomada como ejemplo, contó con 2790 *slots* por ordenar. Al ser un problema de permutación, se planteó un intercambio por pares de *slots*. Sin embargo, explorar un vecindario resultó en un tiempo computacional elevado y la mejora que representaba en el valor de la función objetivo era mínima. Por lo tanto, se decidió separar el total de *slots* en grupos para realizar el intercambio entre estos. En este caso, se usaron cuatro tamaños de grupo: 48, 60, 72 y 84 *slots*.

Matriz y duración tabú

Como bien lo indica el autor en [17], para evitar la búsqueda de combinaciones de intercambio repetidas usadas en el pasado reciente, se definió un número de movimientos prohibidos o tabú. Esto quiere decir que se guardarán en la memoria todos los intercambios compuestos por cualquiera de los pares de grupos más recientes y no podrán realizarse durante cinco iteraciones. Se muestra un ejemplo en la ilustración 3 (*ver ilustración 3*) cuando se tiene un tamaño de grupo de 48 *slots*, el tamaño de la matriz tabú es de 59.

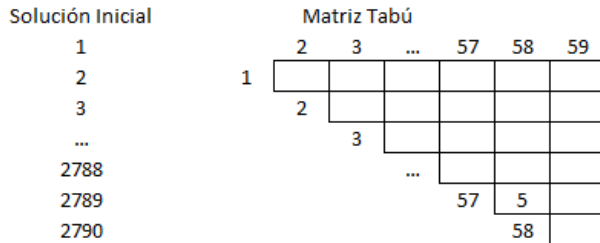


Ilustración 3. Vector de solución inicial y matriz tabú. Elaboración propia.

Criterio de parada

Se escogieron dos criterios de parada para el algoritmo: un número determinado de iteraciones sin mejora de la función objetivo o el tiempo máximo de procesamiento permitido.

Análisis de los resultados obtenidos con los diferentes criterios de anchos de grupos

En la ilustración 4 se muestran los resultados de las pruebas realizadas para los tamaños de los grupos de cambio (*ver ilustración 4*). Tras comparar el tiempo de viaje generado por cada ancho de grupo, se escogió el ancho de 84 como criterio de aplicación de la búsqueda tabú en la comparación de los tiempos de viaje generados por los ordenamientos de las diferentes metaheurísticas, debido a que este generó la mejor solución.

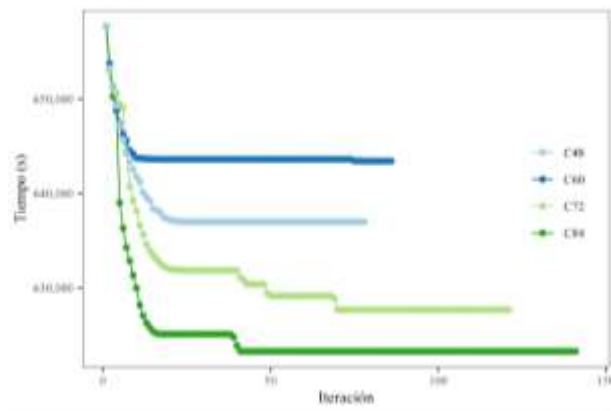


Ilustración 4. Resultados búsqueda tabú. Elaboración propia.

Buscando salir de los óptimos locales, se desarrollaron pruebas adicionales. Cada una de las pruebas se realizó cambiando alguno de los criterios tabú como el aumento del tiempo máximo de procesamiento, el aumento de la cantidad de iteraciones sin mejora y un bloqueo tabú muy grande. Estas pruebas llegaron a los mismos resultados que las primeras elaboradas.

Algoritmo genético

Posteriormente, se implementó el algoritmo genético, el cual es un método adaptativo poblacional, en donde cada individuo representa una posible solución al problema y los individuos evolucionan sobre el principio de selección natural [18].

Codificación

En este caso, la codificación implementada para el cromosoma es de **tipo entero vectorial**. En esta, el vector representa el orden en que deben ser ubicados los SKUs en la zona de *picking*.

Población inicial

En la población inicial del algoritmo genético, se incorporó el ordenamiento de la heurística de ejemplo como un individuo y el resto de individuos correspondieron a ordenamientos aleatorios.

Operadores genéticos

Para el análisis del comportamiento de la metaheurística, los escenarios propuestos en esta sección se corrieron durante el mismo tiempo. Además del tiempo de procesamiento, en todos los escenarios se mantuvieron fijos los siguientes operadores genéticos:

- Escogencia de padres: método de ruleta.
- Probabilidad de ser padre: *fitness* normalizado.
- Cruce: dos puntos.
- Mutación: se realiza un *swap* entre genes, del 3% de la longitud del cromosoma. La probabilidad de mutación es de 1%.

Estrategias de aplicación del algoritmo genético

Se estableció realizar un equilibrio entre intensificar y diversificar el espacio de búsqueda a través del desarrollo de diferentes estrategias de reemplazo de población y manteniendo constantes los demás operadores genéticos.

La primera estrategia de reemplazo realizada consistió en mantener a todos los hijos como la siguiente población, conservando al mejor individuo de la generación anterior. La segunda estrategia de reemplazo es similar a la primera, con la diferencia de que, en esta se conservaron los dos mejores individuos de la generación anterior. Para la tercera estrategia, se implementó la segunda hasta una generación n (en este caso $n = 200$) y, en adelante, el reemplazo de las nuevas poblaciones se hizo conservando a los individuos más aptos y un porcentaje de los menos aptos, entre padres e hijos. Se probaron cinco escenarios del porcentaje de individuos menos aptos por conservar en las poblaciones, estos fueron: 0%, 20%, 50%, 65% y 80%.

La ilustración 5 muestra los resultados del mejor individuo generado hasta cada generación para cada una de las estrategias (*ver ilustración 5*). En esta, se observa que el mejor resultado se obtiene al implementar la tercera estrategia con un 50% de individuos menos aptos. Por tanto, esta estrategia se escogió para la implementación del algoritmo genético en la comparación de los tiempos de viaje generados por los ordenamientos de las diferentes metaheurísticas.

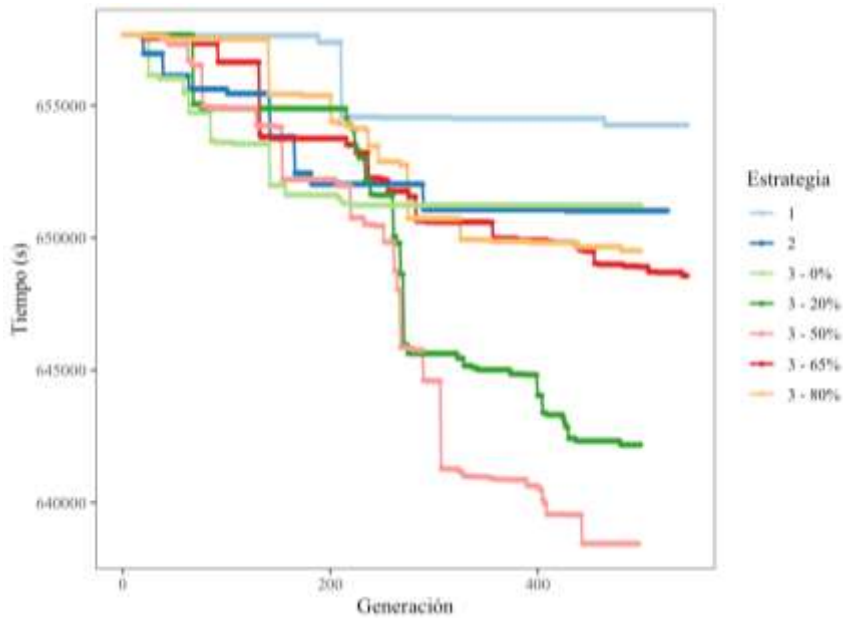


Ilustración 5. Resultados de los mejores individuos en cada estrategia de algoritmo genético. Elaboración propia.

4.1.2.1. Análisis de comparación entre metaheurísticas

Análisis de tiempo de cómputo

Por un lado, el tiempo de cómputo en el tabú varió en función de los anchos de grupos que se probaron. Entre menor fue el tamaño de grupo probado, mayor fue el tiempo de cómputo empleado. Esto se debió a que el tamaño del vecindario fue más grande y, por tanto, probar todas las combinaciones tomó más tiempo. Por otro lado, en el algoritmo genético, el tiempo de cómputo se mantuvo constante para todas las estrategias.

Análisis de la convergencia en las metaheurísticas

Por un lado, la convergencia de la búsqueda tabú se da cuando pasan un número de iteraciones sin que la FO mejore, llegando a óptimos locales. Por otro lado, la convergencia en el algoritmo genético se da cuando en las nuevas poblaciones, el cromosoma de los individuos es igual, es decir, se pierde la diversidad.

En el caso de la metaheurística de búsqueda tabú, los resultados con los diferentes anchos de grupo (*ver ilustración 4*) presentaron una mejora significativa en las iteraciones iniciales, pero luego tendieron a llegar a óptimos locales. Por tanto, para los diferentes anchos de grupo probados, esta metaheurística presentó una convergencia rápida, llegando al 95% de la mejora total aproximadamente en la iteración 20.

En el caso del algoritmo genético, se observó que en las primeras dos estrategias había diversidad en los individuos y ninguna de ellas convergió al final del procesamiento (*ver ilustración 6 y 7*).

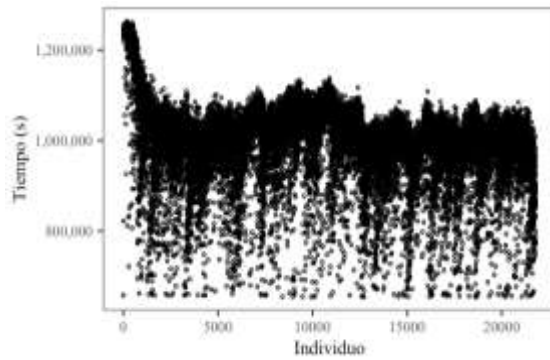


Ilustración 6. Resultados estrategia 1 algoritmo genético. Elaboración propia.

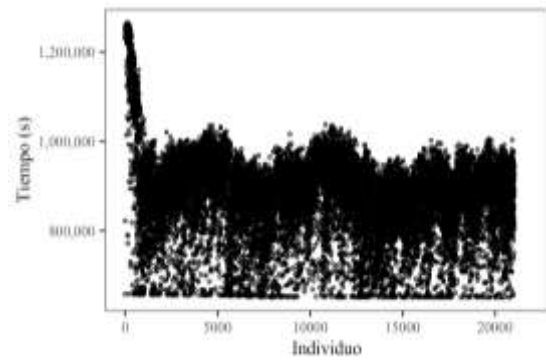


Ilustración 7. Resultados estrategia 2 algoritmo genético. Elaboración propia.

Respecto a la tercera estrategia, se observa que el cambio de reemplazo de la población para todos los escenarios acelera la convergencia. En los escenarios de 0% (ver ilustración 8) y 20% (ver ilustración 9) resalta el cambio en la actualización de población, demostrando que, luego de la generación del cambio, los nuevos individuos tienden a ser similares entre sí y se pierde la diversidad. Para los escenarios de 50% (ver ilustración 10) y 65% (ver ilustración 11), se observó un descenso más lento en la pérdida de diversidad. Por último, para el escenario de 80% se observó que la diversidad en los individuos se mantiene a lo largo del procesamiento (ver ilustración 12).

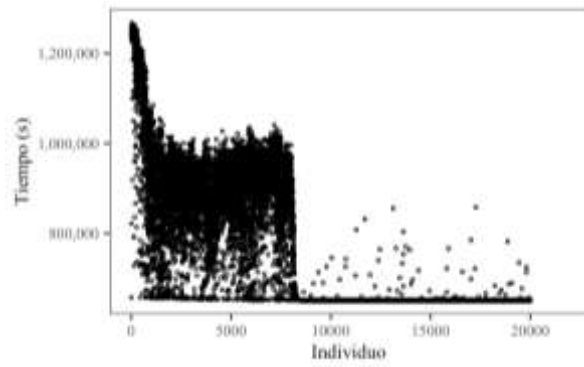


Ilustración 8. Resultados de la estrategia 3 conservando 0% de los menos aptos para el algoritmo genético. Elaboración propia.

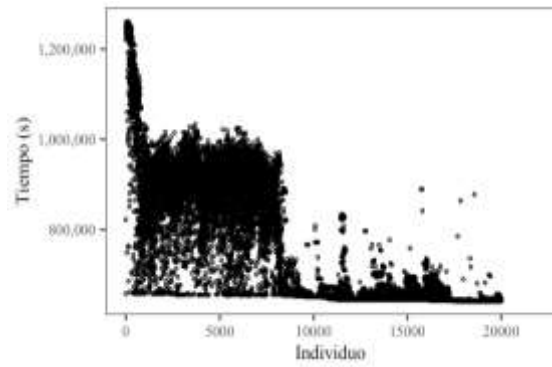


Ilustración 9. Resultados de la estrategia 3 conservando 20% de los menos aptos para el algoritmo genético. Elaboración propia.

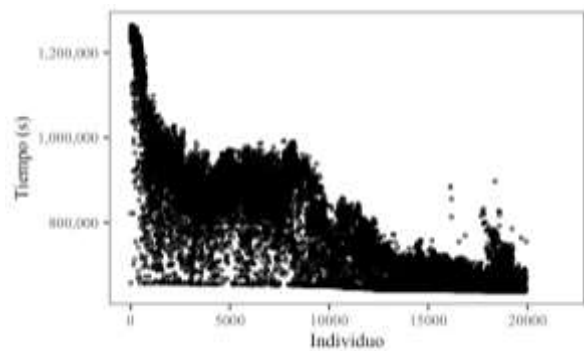


Ilustración 10. Resultados de la estrategia 3 conservando 50% de los menos aptos para el algoritmo genético. Elaboración propia.

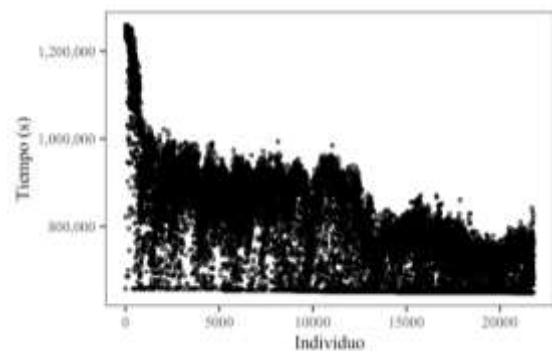


Ilustración 11. Resultados de la estrategia 3 conservando 65% de los menos aptos para el algoritmo genético. Elaboración propia.

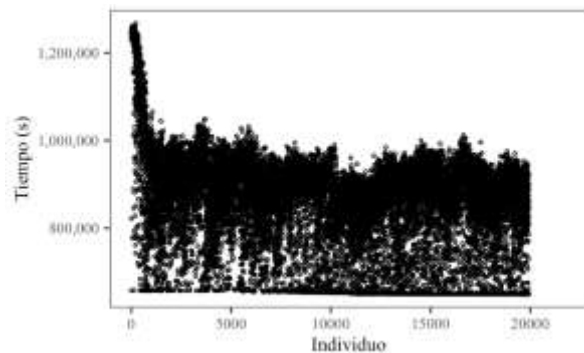


Ilustración 12. Resultados de la estrategia 3 conservando 80% de los menos aptos para el algoritmo genético. Elaboración propia.

De acuerdo con los dos apartados anteriores, se observó que la búsqueda tabú generó una buena solución rápidamente (menos de una hora de procesamiento), pero todas las soluciones cayeron en óptimos locales. Por el contrario, el algoritmo genético no generó una solución tan buena como la del tabú, pero se preservó la diversidad en los individuos para la mayoría de las estrategias. Por lo tanto, para aprovechar la intensificación de la búsqueda tabú y la diversificación en el algoritmo genético, en el siguiente apartado se presenta el uso conjunto de las metaheurísticas.

Combinación de búsqueda tabú y algoritmo genético

Con base en los resultados obtenidos con cada metaheurística, se decidió implementar el mejor ordenamiento generado por el tabú (ancho de 84) como un individuo en el algoritmo genético. Para este último, se usaron los operadores genéticos descritos en la sección del algoritmo genético y, como estrategia de reemplazo de la población, se usó la estrategia que generó la mayor reducción de la FO (estrategia 3 con 50% de individuos menos aptos).

Mediante la combinación de ambas metaheurísticas, se pudo observar que el algoritmo genético ayudó a resolver el problema de óptimos locales que presentaba el tabú (ver ilustración 13).

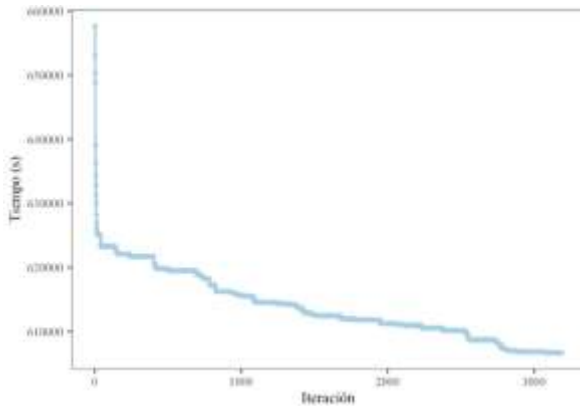


Ilustración 13. Resultado de la implementación de búsqueda tabú y algoritmo genético. Elaboración propia.

Para revisar el aplicativo de ordenamiento por metaheurísticas, ver *Anexo 3*.

4.2. Cantidad de *slots* fijos y porcentaje de *slots* dinámicos

Siguiendo con la metodología de solución, en el segundo paso se presenta un modelo de programación lineal como estrategia para determinar la cantidad de *slots* fijos a asignar a cada SKU. Luego, se presenta el método de ajuste de las cantidades en función del porcentaje de *slots* dinámicos a evaluar.

4.2.1. Modelo de programación lineal

Con el fin de calcular la cantidad de *slots* fijos a asignar a cada SKU, se desarrolló un modelo de programación lineal. Este modelo buscó reducir los reabastos a través de la minimización de unidades por reabastecer de los SKUs durante el periodo mayo-julio.

Se corrió en Gusek (ver *Anexo 4*) el modelo matemático que se expone a continuación:

Conjuntos	
Expresión	Descripción
$I: \{1 \dots I \}$	<i>Conjunto de SKUs</i>
$J: \{1, 2, 3\}$	<i>Conjunto de los meses de demanda a tomar en cuenta (mayo, junio y julio)</i>
Parámetros	
Expresión	Descripción

$D_{ij} \forall i \in I j \in J$	<i>Demanda del SKU $i \in I$ en el mes $j \in J$</i>
Cap	<i>Capacidad de los slots</i>
CT	<i>Cantidad de slots en el piso</i>
$NumNuevos$	<i>Cantidad de slots asignados a referencias nuevas</i>
Variables de decisión	
Expresión	Descripción
$X_{ij} \forall i \in I j \in J$	<i>Cantidad de slots que tiene el SKU $i \in I$ en el mes $j \in J$</i>
$Unidades\ por\ reabastecer_{ij}$ $\forall i \in I j \in J$	<i>Unidades por reabastecer del SKU $i \in I$ en el mes $j \in J$</i>
$Y_i \forall i \in I$	<i>Cantidad ponderada de slots asignados durante los tres meses al SKU $i \in I$</i>

Función Objetivo

$$\min: \sum_{j \in J} \sum_{i \in I} Unidades\ por\ reabastecer_{ij}$$

Restricciones

$$Unidades\ por\ reabastecer_{ij} = D_{ij} + Cap - X_{ij} * Cap, \forall i \in I j \in J \quad (1)$$

$$\sum_{i \in I} X_{ij} \leq CT, \forall j \in J \quad (2)$$

$$Y_i = X_{i1} * 0.2 + X_{i2} * 0.3 + X_{i3} * 0.5, \forall i \in I \quad (3)$$

$$\sum_{i \in I} Y_i \leq (CT - NumNuevos) \quad (4)$$

La función objetivo minimizó el total de unidades por reabastecer durante los tres meses analizados. La restricción (1) evaluó la cantidad de unidades por reabastecer para cada SKU en cada mes analizado, asegurando que se asignara al menos un *slot* a cada SKU. La restricción (2) aseguró que, para cada mes, la cantidad de *slots* asignados no superara la cantidad total de *slots* en el piso. La restricción (3) calculó la cantidad de *slots* a asignar a cada referencia durante el periodo de tres meses, realizando una ponderación que otorgó mayor importancia a la cantidad de *slots* de los meses más recientes. La restricción (4) aseguró que la cantidad total de *slots* para todos los SKUs no superara la cantidad total de *slots* en el piso, teniendo en cuenta la cantidad de referencias nuevas.

Supuesto para la capacidad de slots

En la Torre, cada *slot* tiene una capacidad específica para cada SKU. Sin embargo, debido a que no se contó con la información de la relación capacidad *slot*-SKU, para el desarrollo de la metodología donde fuera necesaria esta información, se hizo el supuesto de que cada *slot* tuvo una capacidad de 100 unidades para cualquier SKU.

4.2.2. Método de ajuste de la cantidad de slots fijos de acuerdo con el porcentaje de dinámicas

Mediante la macro 1 del aplicativo de consolidación del *slotting*, desarrollado en el lenguaje de programación de Excel VBA (ver *Anexo 5*), se ajustó la cantidad de *slots* fijos obtenidos del modelo lineal a diferentes porcentajes de *slots* dinámicos. A partir de esto, se generaron diez escenarios: 0%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45% y 50%. A continuación, la tabla 1 presenta el pseudocódigo del algoritmo (ver *tabla 1*):

Tabla 1. Pseudocódigo macro 1 aplicativo de consolidación de *slotting*. Elaboración propia.

Macro 1	Ajuste cantidad slots
Leer cantidadTotal_slotsPiso	
Leer porcentajeRequerido_slotsDinámicos	
Leer cantidadTotal_slotsFijos_ModeloLineal	
cantidadActual_slotsDinámicos= cantidadTotal_slotsPiso - cantidadTotal_slotsFijos_ModeloLineal	
cantidadRequerida_slotsDinámicos= cantidadTotal_slotsPiso * porcentajeRequerido_slotsDinámicos	
cantidadRequerida_slotsFijos= cantidadTotal_slotsPiso - cantidadRequerida_slotsDinámicos	
ajusteSlots_por_SKU= cantidadRequerida_slotsFijos / cantidadTotal_slotsFijos_ModeloLineal	
Si cantidadRequerida_slotsFijos > cantidadTotal_slotsFijos_ModeloLineal entonces	
Para todo SKU s en SKUs	
Si (cantidadTotal_slotsFijos_ModeloLineal + ajusteSlots_por_SKU) < cantidadRequerida_slotsFijos entonces	
cantidadSlotsFijos(s)= cantidadSlotsFijos(s) + ajusteSlots_por_SKU	
cantidadTotal_slotsFijos_ModeloLineal=cantidadTotal_slotsFijos_ModeloLineal+ajusteSlots_por_SKU	
Fin del condicional	
Fin del Ciclo	
Fin del condicional	
Si cantidadRequerida_slotsFijos < cantidadTotal_slotsFijos_ModeloLineal entonces	
Para todo SKU s en SKUs	
Si cantidadTotal_slotsFijos_ModeloLineal > cantidadRequerida_slotsFijos entonces	
Si cantidadSlotsFijos(s) > 0 entonces	
cantidadSlotsFijos(s)= cantidadSlotsFijos(s) - ajusteSlots_por_SKU	
cantidadSlotsFijos_ModeloLineal=cantidadSlotsFijos_ModeloLineal - ajusteSlots_por_SKU	
Fin del condicional	
Fin del condicional	
Fin del Ciclo	
Fin del condicional	

4.3. Política de mantenimiento

Siguiendo con la metodología de solución, en el paso tres se determinó una política de mantenimiento que actualiza periódicamente la capacidad asignada a cada SKU según su demanda, con el fin de disminuir el tiempo de reabasto. Para ello, se buscó que la capacidad asignada a cada SKU en la Torre, soportara una semana de demanda sin necesidad de realizarle reabasto.

Esta política consta de las variables que se explican a continuación:

- Frecuencia de mantenimiento*: se refiere a la periodicidad con la que se realiza el mantenimiento. Para determinar la frecuencia adecuada de mantenimiento, se probaron diferentes escenarios con frecuencia de una, dos, tres y cuatro semanas.
- Duración del periodo histórico considerado*: se refiere a la cantidad de semanas de demanda histórica que se tienen en cuenta en el momento de mantenimiento. Para determinar la duración adecuada del periodo histórico, se probaron diferentes escenarios que contemplan la demanda de una, dos y cuatro semanas.
- Cantidad máxima de slots a aumentar/disminuir*: se refiere al límite permitido de *slots* a cambiar en cada mantenimiento, tanto de aumento como de disminución. Se probaron diferentes escenarios en los que se varió la cantidad máxima de cambios en 100, 300 y 600 *slots*.

La ecuación (1) que se empleó para calcular la cantidad de *slots* a aumentar o disminuir por cada SKU se explica a continuación:

$$\text{Slots de cambio}_{i,m} = \frac{\text{Promedio cant. dem.}_{ip} - \text{Capacidad slot} * (\text{Fijos}_{i,m-1} + \text{Dinámicos}_{i,m-1})}{\text{Capacidad slot}} \quad (1)$$

Donde:

$\text{Slots de cambio}_{i,m}$: cantidad de *slots* a modificar para el SKU *i* en el mantenimiento *m*.

- Si el valor es positivo, se debe aumentar la cantidad de *slots* para el SKU *i* en la cantidad de unidades que el resultado indique.
- Si el valor es negativo, se debe disminuir la cantidad de *slots* para el SKU *i* en la cantidad de unidades que el resultado indique.
- Si el valor es 0, se mantiene la cantidad de *slots* que el SKU *i* tiene en el mantenimiento *m*.

$\text{Promedio cant. dem.}_{ip}$: promedio de la cantidad semanal demandada del SKU *i* durante el periodo histórico *p* considerado. De acuerdo con el escenario, el periodo *p* puede contemplar la demanda de una, dos o cuatro semanas anteriores al momento que se está evaluando.

$\text{Fijos}_{i,m-1}$: cantidad de *slots* fijos que tiene asignado el SKU *i* en el momento que se está evaluando.

$\text{Dinámicos}_{i,m-1}$: cantidad de *slots* dinámicos que tiene asignado el SKU *i* en el momento que se está evaluando.

Capacidad slot : capacidad de los slots (100 und/slot).

Restricciones del mantenimiento

Debido a que el porcentaje de *slots* dinámicos en la Torre debe permanecer constante, es necesario aumentar y disminuir *slots* en la misma cantidad. Por esto, para realizar ambos procesos se escoge el valor mínimo entre la cantidad total por disminuir y la cantidad total por aumentar, respetando la restricción de cambios máximos.

4.4. Aplicativo de consolidación del slotting

La macro 2 del aplicativo de consolidación del *slotting* (ver *Anexo 5*) se desarrolló para unificar el ordenamiento que disminuyó en mayor medida el tiempo de viaje y cada uno de los diez escenarios de *slots* dinámicos generados en la sección 4.2.2. A continuación, se muestra el pseudocódigo de la macro:

Tabla 2. Pseudocódigo macro 2 aplicativo de consolidación de slotting. Elaboración propia.

Macro 2	Consolidación ordenamiento y cantidad
Leer cantidadRequerida_slotsDinámicos	
Calcular slotsDinámicos_zona	
Calcular slotsFijos_zona	
zona=1	
secuencia=0	
Para todo SKU s en Ordenamiento_SKUs	
Si secuencia+cantidadSlotsFijos(s)<=slotsTotales entonces	
Para cantSlots=1 hasta cantidadSlotsFijos(s)	
secuencia=secuencia+1	
slotsPiso(secuencia)=Ordenamiento_SKUs(s)	
Fin del ciclo	
Si secuencia>slotsFijos_zona Y zona<=8 entonces	
Para dinámicos=1 hasta slotsDinámicos_por_zona	
secuencia=secuencia+1	
slotsPiso(Secuencia)=""	
Fin del ciclo	
zona=zona+1	
slotsFijos_zona=slotsZona*zona-(slotsZona/2)	
Fin del condicional	
Fin del condicional	
Fin del ciclo	

4.5. Panoramas de demanda para evaluar el lead time

Con el fin de analizar el comportamiento del *lead time* de acuerdo con los *slottings* consolidados en el apartado 4.4., se evaluaron tres panoramas de demanda diferentes:

- a) *Demanda real*: se tomaron los pedidos reales de la compañía del periodo agosto-octubre. En esta demanda, el 55% de los SKUs se encuentran en el *slotting* generado por el aplicativo de consolidación, mientras que el 45%, son referencias nuevas que entran a lo largo del trimestre.
- b) *Demandas hipotéticas*: se modificaron los pedidos reales mediante el aumento y la disminución de la variabilidad de las referencias demandadas entre agosto y octubre para generar dos panoramas:
 - i) Alta variabilidad: en esta demanda, el 28% de los SKUs se encuentran en el *slotting* generado por el aplicativo de consolidación, mientras que el 72%, son referencias nuevas que entran a lo largo del trimestre.
 - ii) Baja variabilidad: en esta demanda, el 97% de los SKUs se encuentran en el *slotting* generado por el aplicativo de consolidación, mientras que el 3%, son referencias nuevas que entran a lo largo del trimestre.

Teniendo en cuenta las variables de mantenimiento y los diez *slottings*, se evaluó un total de 360 escenarios para cada panorama de demanda. Adicionalmente, se evaluaron los diez *slottings* sin mantenimiento.

4.6. Aplicativo de simulación del lead time

Con el fin de evaluar la efectividad de cualquier propuesta de *slotting*, se desarrolló un aplicativo en el lenguaje de programación de Excel VBA que calculó el *lead time* de la operación para un piso de la Torre. Este algoritmo simuló políticas actuales de la empresa y políticas de manejo propuestas en este proyecto.

Las políticas simuladas de Totto se refieren a la creación de unidades de trabajo (UTs), la creación de olas de pedido, el proceso de reabasto, el mantenimiento de ubicaciones dinámicas y el recorrido de las rutas de

picking. Mientras que las políticas propuestas en este proyecto son la estrategia de mantenimiento del *slotting* y el manejo de unidades sin cupo.

Aproximaciones de la simulación respecto a la operación del CEDI

- a) *Tiempo por unidades sin cupo*: es el tiempo de penalización en el *lead time* generado por las unidades que no pueden ser recogidas al momento del *picking*, debido a que la Torre se encuentra a su máxima capacidad. Este tiempo corresponde a la duración aproximada de su recolección.

El concepto de unidades sin cupo fue creado específicamente para la simulación del *lead time* en el aplicativo. En la operación real de la empresa, la Torre no llega a su máxima capacidad, debido a que se realizan ajustes de acuerdo con la información de proyección de demanda. Estos ajustes no fueron incluidos en el archivo de simulación porque dependen de diversas variables de la cadena de suministro.

- b) *Tiempo de reabasto*: mediante un análisis de los tiempos estimados de la compañía, se determinó un modelo de regresión lineal para hallar el tiempo de reabasto aproximado en términos de la cantidad de cajas de producto a reabastecer, como se muestra en la ecuación (2). [Con este modelo, se obtuvo un coeficiente de determinación R²=0,5553.](#)

$$y = 0,5647x + 12,603 \tag{2}$$

Donde y es el tiempo de reabasto, x es el número de cajas a reabastecer.

Con base en lo anterior, los tiempos que componen la función objetivo simulada por el aplicativo son los siguientes: tiempo de reabasto, tiempo de viaje, tiempo de *pick* y tiempo por unidades sin cupo.

La tabla 3 que presenta la lógica para evaluar la función objetivo de este aplicativo, se muestra a continuación (*ver Tabla 3*):

Tabla 3. Subrutina principal del aplicativo de simulación de lead time. Elaboración propia.

Aplicativo	Simulación del Lead Time
	(1) llamar_Leer
	(2) llamar_crearUT
	(3) llamar_creacionOlas
	(4) llamar_correrCalculo
	(5) llamar_imprimirTiempos

En la subrutina (1), se leen los parámetros de entrada. Luego, en la subrutina (2) y (3), se lleva a cabo la creación de las unidades de trabajo y olas de pedidos, explicadas en la sección de justificación y contexto del problema. Posteriormente, en la subrutina (4) se desarrollan los procesos más robustos de la operación en la Torre como el alistamiento de pedidos siguiendo las rutas preestablecidas por la empresa, el reabasto y las políticas de mantenimiento del *slotting*. Para ver más detalles sobre el pseudocódigo, ver *Anexo 6* y sobre el aplicativo, ver *Anexo 7*.

5. Resultados

En esta sección, se exponen los resultados del tiempo de viaje generado por los ordenamientos heurísticos y metaheurísticos y los resultados del *lead time* obtenidos de la evaluación de escenarios generados en el apartado 4.5.

5.1. Resultados de ordenamiento

5.1.1. Resultados del tiempo de viaje obtenido por las heurísticas

La heurística cuyo ordenamiento generó el menor tiempo de viaje fue la de correlaciones, seguida de la multicriterio, la clasificación ABC (Totto) y, por último, la aleatoria. Respecto al ordenamiento que maneja actualmente la compañía, el ordenamiento por correlaciones redujo el tiempo de viaje en 27% (ver ilustración 14).

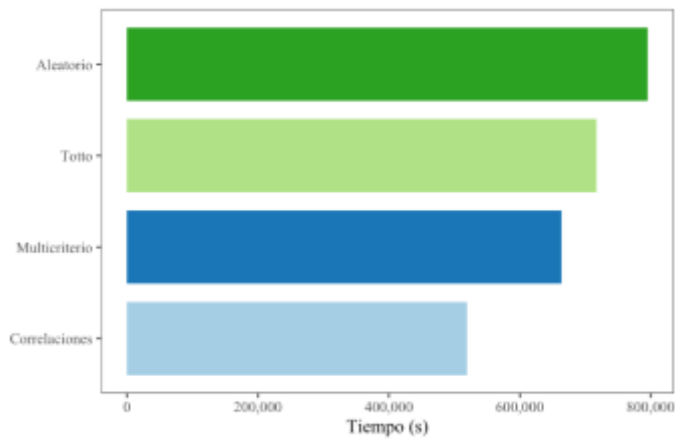


Ilustración 14. Comparación de tiempos de viaje obtenidos por cada heurística. Elaboración propia.

5.1.2. Resultados del tiempo de viaje obtenido por las metaheurísticas

A partir de la solución por correlaciones, se exponen los resultados obtenidos con las metaheurísticas de búsqueda tabú, algoritmo genético y la combinación de las dos. Esta última, fue la que redujo en mayor cantidad la FO, donde la búsqueda tabú disminuyó la función objetivo en 5.3%, mientras que el algoritmo genético, partiendo desde la búsqueda tabú, la disminuyó en 2.9% para un total de 7.75% de reducción en el tiempo de viaje (ver ilustración 15).

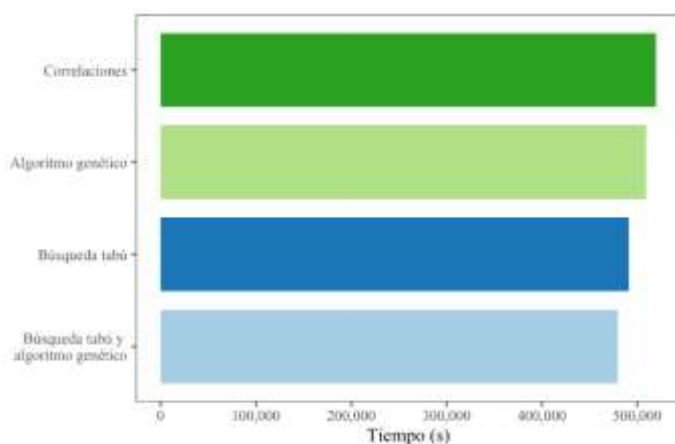


Ilustración 15. Comparación de tiempo de viaje generado por las metaheurísticas y la mejor heurística. Elaboración propia.

5.2. Resultados de panoramas de demanda

En esta sección se analizaron los resultados obtenidos de los escenarios de *slotting* desarrollados con el aplicativo de la sección 4.4 y la política de mantenimiento propuesta.

5.2.1. Resultados demanda real

Impacto en el lead time con el uso de la política de mantenimiento

Para todos los escenarios de porcentaje de *slots* dinámicos, siempre existió una mejor solución implementando una política de mantenimiento a lo largo del horizonte de tiempo (ver *ilustración 16*). Así pues, el *lead time* para los escenarios con mantenimiento, se redujo en promedio 36% respecto a los escenarios sin mantenimiento.

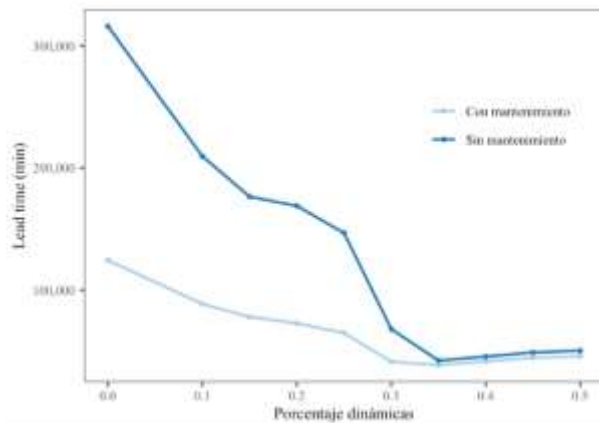


Ilustración 16. Lead time en función de porcentaje de slots dinámicos con mantenimiento y sin mantenimiento. Elaboración propia.

Dado que el mantenimiento generó una mejora significativa en el *lead time*, los resultados mostrados a continuación son los escenarios en los que se implementó la política de mantenimiento.

Comportamiento del lead time respecto al porcentaje de slots

En la *ilustración 17*, se puede observar el comportamiento promedio, mínimo y máximo del *lead time* para cada porcentaje de *slots* dinámicos (ver *ilustración 17*). A partir del 30% de *slots* dinámicos se obtuvieron los menores tiempos, siendo 35% el porcentaje que generó la mejor respuesta.

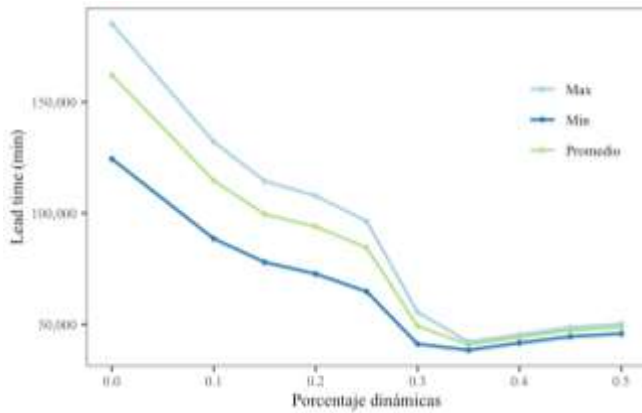


Ilustración 17. Lead time en función de porcentaje de slots dinámicos con mantenimiento. Elaboración propia.

Análisis de los componentes del lead time para 35% de dinámicas

Con respecto a la variable de frecuencia de mantenimiento se puede observar que los mejores resultados en el lead time se obtuvieron realizando el mantenimiento semanalmente (ver ilustración 18).

El mejor lead time se obtuvo al utilizar un 35% de dinámicas, un máximo de 600 cambios, una semana de historial y una frecuencia semanal (ver ilustración 18). De este, el tiempo de reabasto representó un 59% del tiempo total, el tiempo de viaje un 11% y el tiempo de pick un 30%.

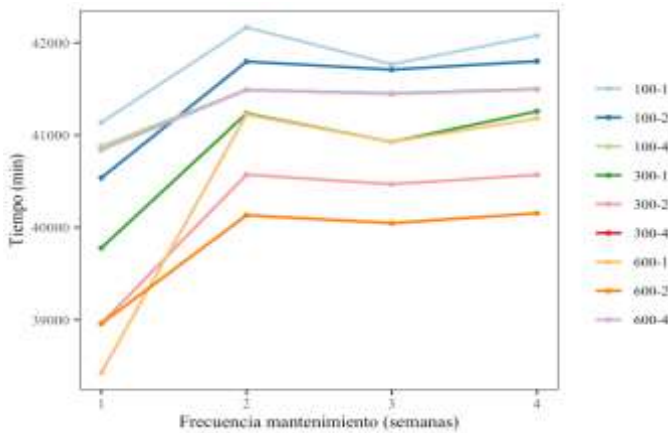


Ilustración 18. Lead time con las variables de mantenimiento y 35% de slots dinámicos. Elaboración propia.

En la ilustración 18, los valores de la leyenda están conformados por dos números separados por un guión. El número de la izquierda representa la cantidad máxima de cambios permitidos en el mantenimiento y el de la derecha, la cantidad de semanas de historial.

El lead time de la operación se redujo al realizar un mantenimiento semanal, debido a que se llevó a cabo una actualización más continua del slotting a los requerimientos de la demanda, minimizando el número de reabastos y, por tanto, el tiempo de reabasto (proceso con mayor porcentaje del lead time) (ver ilustración 19). Por el contrario, en el tiempo de viaje, se observaron mayores tiempos debido a que el ordenamiento inicial se vio afectado por los cambios frecuentes (ver ilustración 20).

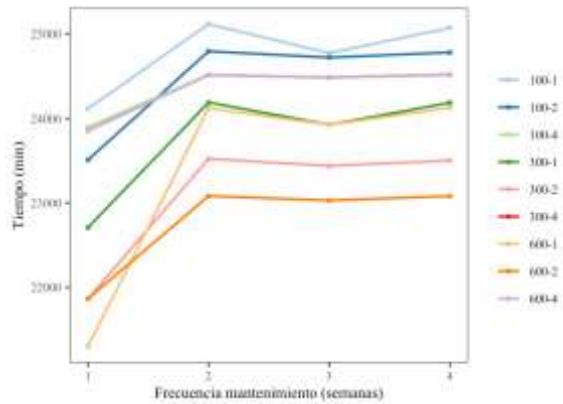


Ilustración 19. Tiempo de reabasto con las variables de mantenimiento y 35% de slots dinámicos. Elaboración propia.

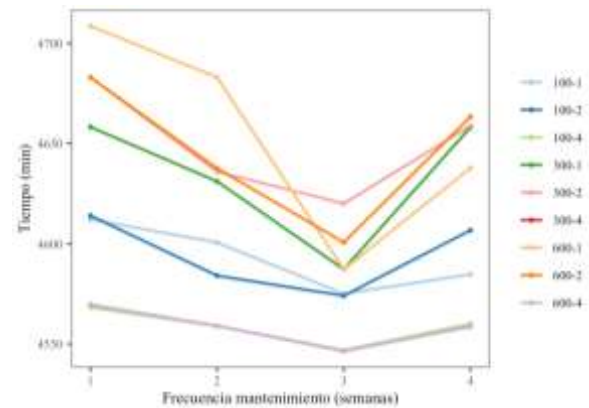


Ilustración 20. Tiempo de viaje con las variables de mantenimiento y 35% de slots dinámicos. Elaboración propia.

5.2.2. Demandas hipotéticas: alta y baja variabilidad

Comportamiento del lead time

Con una alta variabilidad en la demanda, se obtuvo que los mejores *lead time* se generaron a partir del 25% de *slots* dinámicos mientras que, con una baja variabilidad, se observó que los mejores tiempos se obtuvieron con porcentajes entre el 15% y 35%. En las ilustraciones 21 y 22, se pueden observar los tiempos promedio, el tiempo máximo y mínimo con respecto a cada escenario de *slots* dinámicos (ver ilustraciones 21 y 22).

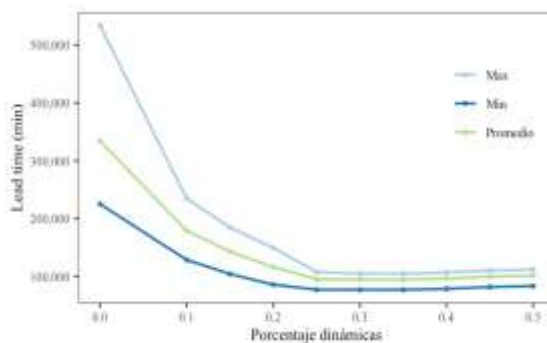


Ilustración 21. Lead time en función de porcentaje de slots dinámicos con mantenimiento para un panorama con alta variabilidad. Elaboración propia.

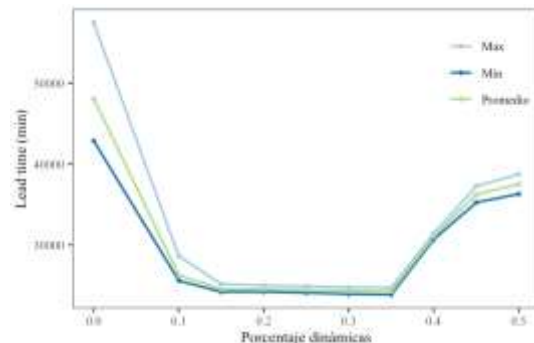


Ilustración 22. Lead time en función de porcentaje de slots dinámicos con mantenimiento para un panorama con baja variabilidad. Elaboración propia.

Composición del lead time

Con respecto a la descomposición del *lead time*, se obtuvo que para una alta variabilidad en promedio, el reabasto tiene el mayor peso entre todos los tiempos, con un 55% y, en todos los escenarios, se presentó el tiempo por unidades sin cupo. Por el contrario, para una baja variabilidad, el tiempo de reabasto perdió peso, ocupando el 30% del *lead time*, y el tiempo de *picking* fue el más importante con un peso de 45%. Además, solo se presentó tiempo por unidades sin cupo en los escenarios con 0% y 10% de *slots* dinámicos (ver ilustración 23 y 24).

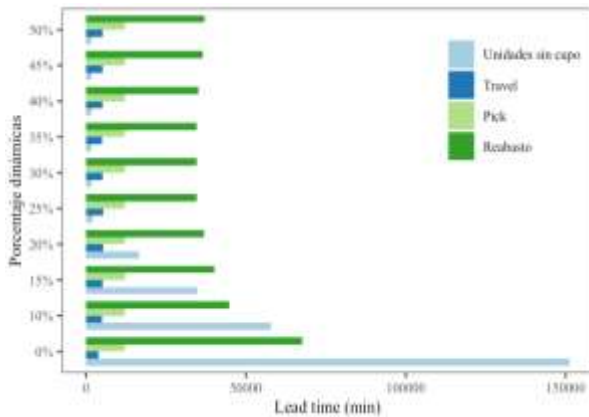


Ilustración 23. Composición del lead time en función del porcentaje de slots dinámicos para un panorama con alta variabilidad. Elaboración propia.

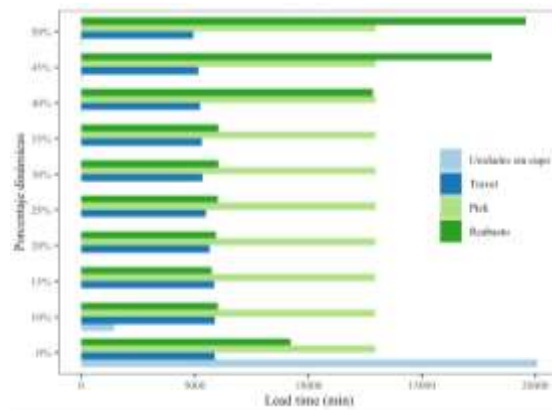


Ilustración 24. Composición del lead time en función del porcentaje de slots dinámicos para un panorama con baja variabilidad. Elaboración propia.

Mejor lead time en panoramas hipotéticos de demanda

Para los dos panoramas de demanda, se obtuvo el mejor *lead time* en el escenario de 30% de *slots* dinámicos, un mantenimiento semanal y una semana de historial de demanda, con la diferencia de que, para una alta variabilidad, el mejor tiempo se dio ejecutando un máximo de 600 cambios en el *slotting* y para una baja variabilidad, un máximo de 100 cambios.

6. Impacto del proyecto

Debido a que no se cuenta con los tiempos reales de reabasto y viaje de Totto, para poder realizar la medición del impacto de la metodología propuesta, se calculó el tiempo teórico de la operación de la compañía a través del archivo de simulación de *lead time*, desarrollado en la sección 4.6.

Comparando los tiempos teóricos de la operación generados por la política de la empresa y la política propuesta, se obtuvo un 55% de mejora en el *lead time*. Sin embargo, cabe resaltar que, de la política de Totto, el tiempo por unidades sin cupo representa un 41% del tiempo total y en la operación real de la empresa no existe este concepto. Por tanto, considerando únicamente los tiempos que sí maneja la empresa, se obtuvo una mejora del 29% del *lead time*. Esta mejora provino en su mayoría del tiempo de reabasto, como se ve en la ilustración 25, donde el tiempo de reabasto se redujo 33%, el tiempo de *pick* se mantuvo muy similar y el tiempo de viaje se redujo 18%.

Por un lado, el impacto en el tiempo de reabasto de la política propuesta, provino de la asignación de la cantidad de *slots* para cada SKU, generada por el modelo matemático y su ajuste a un porcentaje de *slots* dinámicos, además de la realización de un mantenimiento al *slotting* a través del tiempo. Por otro lado, el impacto en el tiempo de viaje provino del ordenamiento del método propuesto, obtenido de la combinación de las metaheurísticas búsqueda tabú y algoritmo genético sobre una heurística inicial de correlaciones.

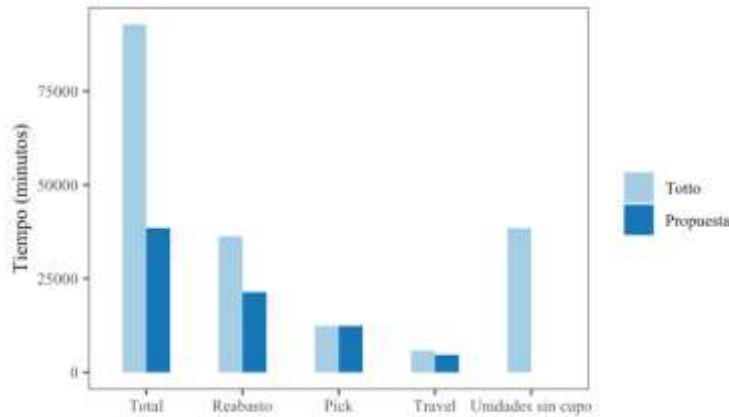


Ilustración 25. Impacto del método propuesto en el lead time de Totto. Elaboración propia.

7. Conclusiones y recomendaciones

Para minimizar el *lead time* de la operación en el centro de distribución Tittán Totto, este proyecto se centró en diseñar un método de asignación de ubicaciones compuesto por tres procedimientos: por un lado, el ordenamiento de los SKUs para reducir el tiempo de alistamiento de pedidos y por otro lado, la cantidad de *slots* fijos a asignar a cada SKU en función del porcentaje de *slots* dinámicos y la política de mantenimiento, para reducir el tiempo de reabasto.

El ordenamiento de los SKUs se obtuvo a partir de la evaluación del tiempo de viaje (componente del tiempo de alistamiento de pedidos) generado por diferentes heurísticas y metaheurísticas. El mejor tiempo se logró con la implementación de una estrategia que partió del ordenamiento de correlaciones y se mejoró con el uso conjunto de las metaheurísticas búsqueda tabú y algoritmo genético. En la evaluación con demanda histórica, esta estrategia redujo el tiempo de viaje en un 34% respecto al ordenamiento de Totto.

La cantidad de *slots* a asignar a cada SKU se obtuvo mediante un modelo de programación lineal y su ajuste a diferentes porcentajes de *slots* dinámicos. La evaluación de cada combinación se realizó respecto al tiempo de reabasto. El mejor tiempo resultó del 35% de *slots* dinámicos, disminuyendo el tiempo de reabasto en un 33% respecto al tiempo de la compañía.

En cuanto a la política de mantenimiento, para todos los escenarios evaluados, siempre se consiguieron mejores tiempos de reabasto cuando esta fue implementada. Con respecto a la mejor solución obtenida sin mantenimiento (escenario con 35% de *slots* dinámicos), el tiempo de reabasto se redujo en un 9%. La combinación de variables de la política de mantenimiento propuesta que minimizó el *lead time* son las siguientes: 35% de *slots* dinámicos, mantenimiento semanal, un máximo de 600 cambios por mantenimiento y una semana de historial en la demanda.

Debido a que el reabasto representa un alto porcentaje del tiempo en el *lead time* de la operación, se recomienda realizar un estudio del desarrollo de este proceso en la compañía.

Debido al impacto que genera el mantenimiento en el *lead time*, se recomienda realizar una investigación detallada sobre las estrategias de mantenimiento para actualizar periódicamente la cantidad de *slots* a la demanda y que además permita mantener el ordenamiento, con el fin de no impactar negativamente el tiempo de viaje.

Referencias

- [1] J. Poornima, A. Azin and R. S. Berntsson, "Impact of usability on process lead-time in information systems: A case study," *The Journal of Systems and Software*, vol. 148, no. 1, pp. 148-169, 2019.
- [2] A. Call, *Reduction of travel distances through slotting, location arrangement and optimized order routing*, Illinois: Northern Illinois University, 2012.
- [3] M. Bataineh, *Minimizing travel distance during warehouse order picking considering congestion effect*, New York: Binghamton University, 2017.
- [4] N. M. Klein, C. M. Klein and J. S. Noble, "Intelligent Location Design for SKU Storage in a Warehouse," in *IIE Annual Conference and Expo*, Nashville, 2015, pp. 546-555.
- [5] C. G. Petersen, C. Siu and D. R. Heiser, "Improving order picking performance utilizing slotting and golden zone storage," *International Journal of Operations & Production Management*, vol. 25, pp. 997-1012, 2005.
- [6] B. Trebilcock, "Resolve to reslot your warehouse," *Modern Materials Handling*, vol. 66, no. 55, pp. 24-28, 2011.
- [7] J. Li, M. Moghaddam and S. Y. Nof, "Dynamic storage assignment with product affinity and ABC classification—a case study," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 9-12, pp. 2179-2194, 2016.
- [8] P. Damodaran, S. Koli, K. Srihari and B. Kimler, "Optimal Product Slotting," in *IIE Annual Conference*, Norcross, 2005, pp. 1-6.
- [9] E. Frazelle and G. Sharp, in *Correlated assignment strategy can improve any order-picking operation.*, 1989, pp. 33-39.
- [10] R.-Q. Zhanga , M. Wanga and X. Pan, "New model of the storage location assignment problem considering demand correlation pattern," *Computers & Industrial Engineering*, vol. 129, no. 1, pp. 210-219, 2019.
- [11] C. H. Glock, P. Kübler and T. Bauernhansl, "A new iterative method for solving the joint dynamic storage location assignment, order batching and picker routing problem in manual picker-to-parts warehouses," *Computers & Industrial Engineering*, vol. 147, no. 1, 2020.
- [12] F. Guerriero, R. Musmanno, O. Pisacane and F. Rende, "A mathematical model for the Multi-Levels Product Allocation Problem in a warehouse with compatibility constraints," *Applied Mathematical Modelling*, vol. 37, no. 6, pp. 4385-4398, 2013.
- [13] R. Koster and W. Delfmann, "Supply chain management," Copenhagen Bussiness School Press, pp. 199-201.

- [14] Y. Zhang, "Correlated Storage Assignment Strategy to reduce Travel Distance in Order Picking," in *Proceedings of the 7th IFAC Conference on Management and Control of Production and Logistics*, Bremen, 2016.
- [15] E. Bottani, M. Cecconi, G. Vignali and R. Montanari, "Optimisation of storage allocation in order picking operations through a genetic algorithm," *International Journal of Logistics: Research and Applications*, vol. 15, no. 2, pp. 127-146, 2012.
- [16] L. Yingde and J. S. Smith, "Dynamic slotting optimization based on SKUs correlations in a zone-based wave-picking system," 12th IMHRC Proceedings , Gardanne, France, 2012.
- [17] B. M. Batista and F. Glover, "Introducción a la Búsqueda Tabú," pp. 2-4.
- [18] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," *Addison-Wesley, Reading, MA*, 1989.