



Facultad de Economía y Negocios
Escuela de Ingeniería Informática Empresarial

DISEÑO Y DESARROLLO DE UN ALGORITMO
HEURÍSTICO PARA LA OPTIMIZACIÓN DE CAJAS
SOBRE PALLETS PARA ALIMENTOS DEL
PEDREGAL

Autores:

Anaís Gaete Veloso
Gonzalo Basualto Ulloa

Profesor guía:

Luis Eduardo Canales

Proyecto de Memoria para optar al título de INGENIERO INFORMÁTICO
EMPRESARIAL.

TALCA – CHILE

2022

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su unidad de procesos técnicos certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Talca, 2023

Agradecimientos

“Le quiero agradecer el apoyo emocional a mis hermanas, padre, madre, abuelos y abuelas, porque fueron mis soportes para no darme por vencido. Sin dudar, tenemos una inspiración recíproca y estuvieron para mí en todo momento. También, quiero agradecer a mi mejor amiga y compañera de tesis Anaís, ya que estuvimos unidos durante 5 años y nunca nos separamos en las adversidades, no tengo dudas de que su futuro será exitoso.”

Gonzalo Antonio Basualto Ulloa

“Quiero agradecer a mis padres y hermano por ser un pilar fundamental en este proceso, apoyarme a cada paso que daba y jamás dejarme sola, todo esto es por y para ustedes. A mi compañero y mejor amigo Gonzalo, porque ningún paso de este viaje lo podría haber dado sin ti, no podría haber elegido a un mejor compañero para recorrer este camino”

Anaís Gaete Veloso

A nuestros amigos Alfonso, Arlette, Alejandra y Sergio, por siempre estar para reconfortarnos, darnos gratos momentos y aprovechar cada oportunidad junto a nosotros. No duden que siempre los vamos a apoyar.

Agradecer a nuestro profesor guía Eduardo Canales, quien nos enseñó lo más importante de ser un buen profesional, nos motivó a seguir siendo las personas que somos y siempre tuvo una palabra de aliento para motivarnos a seguir perseverando. Gracias por enseñarnos durante estos 5 años que hay que dar lo mejor de sí, como docente, como profesional, pero, sobre todo, como persona.

“Arrímate a buen árbol, pues buena sombra te dará.”

Índice de Contenido

Resumen	7
Abstract	8
Introducción.....	9
Objetivos.....	10
Marco Teórico	12
1. Sistemas de información	12
1.1. Tecnologías de la información.....	13
1.2. Sistemas de información logística	14
2. Operaciones logísticas	14
2.1. Costos logísticos	15
3. Algoritmo Heurístico	16
4. Optimización.....	17
4.1. Pyomo	18
5. Problema de carga de contenedores (CLP / PCC)	19
Metodología	23
1. Bosquejo de descripción.....	24
2. Planteamiento de la problemática	26
3. Modelo Matemático	26
4. Algoritmo Heurístico	27
5. Metodología de la evaluación de eficiencia operacional y estimación de costos asociados.....	29
Presentación y análisis de los resultados	30

1. Desarrollo del modelo matemático	30
1.1. Parámetros	30
1.2. Variables	31
1.3. Función objetivo	31
1.4. Restricciones	32
2. Desarrollo de las diferentes versiones del programa.....	36
2.1. Simulación de la ubicación de cajas por nivel sobre el pallet.	36
2.2. Primera versión.....	36
2.3. Segunda versión	38
2.4. Tercera versión.....	41
2.5. Versión Final.....	42
3. Desarrollo de la evaluación de eficiencia operacional	45
3.1. Estimación de costos asociados.....	46
Conclusiones.....	48
Bibliografía.....	50
Anexos	54
1. Anexo 1 – Especificación de requerimientos obtenidos	54

Índice de Cuadros

Cuadro 1 – Tabla comparativa de tiempos	45
Cuadro 2 - Tabla comparativa de volumen.....	46

Índice de Figuras

Figura 1 – Dimensiones de las cajas	20
Figura 2 – Seis orientaciones de elementos ortogonales.....	20
Figura 3 – Iteraciones metodología de desarrollo incremental.....	23
Figura 4 – Caso de uso – Palletizar pedido	25
Figura 5 – Caso de uso – Generar tarja	25
Figura 6 – Caso de uso – Armado de pallet	26
Figura 7 – Diagrama de restricción de cajas dentro del pallet	33
Figura 8 – Diagrama de restricción de traslape	35
Figura 9 – Diagrama de posicionamiento de cajas sobre pallet	36
Figura 10 – Ejemplificación de cajas posicionadas en plano cartesiano.....	40
Figura 11– Ejemplificación de cajas posicionadas en plano cartesiano con nuevo método.....	44

Resumen

Uno de los principales problemas en las operaciones logísticas de una empresa de distribución de productos, es la ubicación óptima de cajas sobre pallets. Alimentos del Pedregal, es una empresa que cuenta con más de 40 salas de ventas distribuidas entre Copiapó y Puerto Montt, sin embargo, carecen de un sistema que permita realizar el armado de pallets de manera rápida y óptima. Actualmente utilizan MS Excel y la experiencia de los trabajadores para la organización de las cajas. En esta dirección, se propone un algoritmo heurístico que realice la ubicación óptima de un conjunto de cajas en el pallet, los cuales pertenecen a sus respectivas notas de ventas, considerando las restricciones y necesidades propias de la empresa.

Las conclusiones de este proyecto indican que el diseño y desarrollo de un algoritmo heurístico para la optimización de cajas sobre pallets ayuda a reducir considerablemente los tiempos de armado de pallets en la empresa Alimentos del Pedregal, permitiendo reducir los costos asociados. Además, nuestros intereses para trabajos futuros relacionados a la problemática están enfocados en un output con un mejor modelado gráfico, que represente más a la realidad, idealmente modelado 3D.

Abstract

One of the main problems in the logistics operations of a product distribution company is the optimal placement of boxes on pallets. Alimentos del Pedregal is a company that has more than 40 sales rooms distributed between Copiapó and Puerto Montt, however, they lack a system that allows them to assemble pallets quickly and optimally. Currently they use MS Excel and the experience of the workers for the organization of the boxes. In this direction, a heuristic algorithm is proposed to perform the optimal placement of a set of boxes on the pallet, which belong to their respective sales notes, considering the restrictions and needs of the company.

The conclusions of this project indicate that the design and development of a heuristic algorithm for the optimization of boxes on pallets helps to considerably reduce pallet assembly times in the company Alimentos del Pedregal, allowing to reduce the associated costs. In addition, our interests for future work related to the problem are focused on an output with a better graphical modeling, which represents more to reality, ideally 3D modeling.

Introducción

Existe una gran cantidad de trabajos en la literatura respecto a la optimización de espacios en contenedores y/o pallets, algunos son García-Cáceres, Vega-Meja, & Caballero-Villalobos (2011), Dell'Amico & Magnani (2021) y Zhu, Oon, Lim, & Weng (2012), sin embargo, estos cuentan con sus respectivas especificaciones, condiciones, restricciones y necesidades, por lo que no hay una única solución que sirva para todas las empresas con procesos logísticos de distribución, esto se debe a que cada organización tiene requerimientos particulares, como por ejemplo los trabajos realizados por Gzara, Elhedhli, & Yildiz (2020), Baldi, Perboli, & Tadei (2012), (Alonso, Alvarez-Valdes, Iori, Parreño, & Tamarit (2017).

Actualmente, Alimentos del Pedregal cuenta con una herramienta Excel, que no considera sus especificaciones, y con la que realizan la distribución de cajas sobre pallets, esto implica aumento de costos debido al uso ineficiente de recursos, como tiempo, espacio, personal humano, transportes, entre otros.

Para una empresa que cuenta con más de 40 salas de ventas distribuidas entre Copiapó y Puerto Montt, es necesaria la implementación de un sistema que permita utilizar de manera eficiente los recursos antes mencionados. A raíz de esto, se propone y desarrolla un algoritmo heurístico que permite optimizar el tiempo de respuesta y la organización de las cajas sobre los pallets tomando en consideración distintas variables, y así entregar una respuesta en tiempo real. Estas variables son: la cantidad de cajas que se quieran organizar en el pallet; la categoría que estas tengan, ya que pueden ser productos delicados; el peso de las cajas, para que las más livianas queden arriba y no aplasten a las otras; dimensiones de las cajas y del camión, para saber cómo organizarlos de manera que ambos espacios sean utilizados más eficientemente.

Dentro de este proyecto se trabajan dos conceptos fundamentales: la optimización, que es uno de los factores más importantes para el éxito de las organizaciones modernas, puesto que permiten mejorar la logística y disminuir el uso de recursos; por otro lado, la heurística que según Silva et al. (2020), utiliza habitualmente pocos recursos para la toma de decisiones, ya que con la gran cantidad de información que se tiene, se busca el camino más fácil para escoger la solución más adecuada en entornos complejos.

Bajo las definiciones presentadas dentro de esta tesis, se puede decir que los algoritmos heurísticos son mucho más rápidos que los algoritmos deterministas, y es debido a esto que, para este proyecto, se decide desarrollar un algoritmo heurístico, el cual tiene un enfoque menos exhaustivo, pero más eficiente, en comparación al algoritmo determinista. Para el desarrollo de este algoritmo se utiliza la metodología de desarrollo incremental.

Objetivos

Objetivo General

Desarrollar un algoritmo heurístico para optimizar el tiempo de respuesta y la organización de las cajas sobre pallets en tiempo real, según variables como cantidad de cajas, categorías, peso, dimensiones de las cajas y dimensiones del camión.

Objetivos Específicos

- Realizar un estudio del arte del problema de la organización de las cajas en los pallets.
- Evaluar soluciones previas de características similares, para ajustar el algoritmo.

- Diseñar y desarrollar una propuesta de algoritmo, para realizar pruebas y llegar a la mejor solución.
- Medir tiempo de respuesta y costos asociados, previos al desarrollo del algoritmo y comparar con medición post desarrollo.

Marco Teórico

Dentro de este capítulo, se abordan y explican conceptos fundamentales para la comprensión del proyecto, y así generar una contextualización del desarrollo del algoritmo.

Como se mencionó anteriormente, este proyecto busca elaborar un algoritmo heurístico, para la optimización, tanto de tiempo, como espacio, de cajas sobre pallets, para la empresa Alimentos del Pedregal, lo que permitirá un mejor aprovechamiento de espacio y probable reducción de costos.

1. Sistemas de información

Un sistema de información, según Varajão, Trigo, Pereira, & Moura (2022), es: “Una combinación de agentes inteligentes (humanos y/o artificiales), procesos y tecnologías de la información (hardware, software e infraestructura) relacionados con la difusión y uso de datos, información y conocimiento en una organización” (p.63). Esto quiere decir que un SI se utiliza para procesar, recolectar y almacenar información, lo cual favorece a las empresas en diversos aspectos del negocio, como por ejemplo la toma de decisiones, la medición de desempeños, entre otros.

Para que un SI sea considerado óptimo, debe contar con tres principales características: ser compatible con otros sistemas, contar con tecnología avanzada y que tenga una capacidad de apoyo, que permita el manejo de información importante, por lo que se debe tener un conocimiento base, para poder administrar la información de todos los sistemas que

interactúan (Cordero Balind, Torres Argüelles, Hernández Gómez, & Ibarra Mejía, 2015). Es por esto que el algoritmo heurístico propuesto para el proyecto se realiza con especial atención a las necesidades de la empresa, para que se integre de buena forma y apoye efectivamente a la optimización de la logística de distribución. Cabe destacar que el algoritmo heurístico que se ha desarrollado no es un sistema de información, pero sí ayuda a agilizar procesos dentro de los SI.

1.1. Tecnologías de la información

Las tecnologías de la información entregan diversos beneficios a las organizaciones, como reducir costos, optimizar procesos, ayuda en la toma de decisiones, entre otros, además, según Szymonik (2012) actualmente las tecnologías de la información son necesarias en cualquier área de gestión de las organizaciones, por ejemplo, el área logística, sin embargo, estas conllevan costos y dificultades que los gerentes deben estar dispuestos a manejar.

En base a lo leído, se pueden entender las tecnologías de la información como uno de los factores base para el desarrollo e implementación de los sistemas de información y/o sistemas de información logísticos modernos, aportando de manera transversal a las necesidades de las organizaciones.

Existen variados tipos de sistemas de información, como por ejemplo sistemas de colaboración empresarial, de información de gestión, de información estratégica, etc. Sin embargo, para efectos de esta investigación el enfoque está en los sistemas de información logística.

1.2. Sistemas de información logística

Las empresas solían utilizar la información con propósitos logísticos para facilitar la toma de decisiones, pudiendo compartir información adecuada y ahorrando recursos, lo que lleva a operaciones más eficientes, obteniendo una gran cantidad de beneficios. Por esto se pensó en un Sistema de información logística, que sea fácil de entender, para permitir la comunicación entre las distintas áreas de la organización, como en el personal de la cadena de suministros y así, poder administrar y comprender fácilmente esta información (Ballou, 2004). Además, Andrejic, Milenkovic, & Sokolovic (2010) mencionan que los SI logística son usados incluso en las fuerzas militares y son casi indispensables para tomar decisiones y reducir costos en el área logística de las distintas instituciones en la actualidad.

Basándose en las fuentes, se puede concluir que el principal objetivo de los sistemas de información logísticos es facilitar tanto la toma de decisiones, como el manejo de los distintos procesos dentro de la cadena de suministros, obteniendo un mayor beneficio con un menor esfuerzo y el desarrollo de este proyecto está estrechamente relacionado a aportar a la mejora de estos procesos.

2. Operaciones logísticas

Según Carro y González (2013), la cadena de suministros es muy importante dentro de las empresas y parte importante de esta son las operaciones logísticas, puesto que son las actividades del área logística que

mantiene satisfecha gran parte de la cadena de suministros. También, Anca (2019), menciona que la innovación en técnicas y uso de tecnologías, respecto a actividades relacionadas al área logística, surgieron de la necesidad de las empresas de comenzar a reducir costos, mejorar calidad de productos y tener clientes satisfechos, entre otros.

Basado en lo anterior, se puede decir que, dentro de la cadena de suministros, para desarrollar un correcto funcionamiento en la empresa, se encuentran las operaciones logísticas, las que cada vez van tomando mayor relevancia y entregando nuevas oportunidades de crecimiento económico en las organizaciones. En esta misma línea, Rimiené & Grundey (2007) señalan que, anteriormente las acciones de las operaciones logísticas estaban enfocadas comúnmente a tareas simples, pero con el avance de las TI, usando nuevos SI y tecnologías más recientes, se han vuelto más complejos, completos y útiles los procesos.

2.1. Costos logísticos

Según Leitão, Silva, Silva, & Brisola (2020), “Los costos logísticos son sumamente relevantes para la sobrevivencia de las organizaciones, por lo tanto, necesitan ser conocidos y monitoreados para lograr el éxito en cualquier negocio” (p.399). Dentro de los costos logísticos, se encuentran todos los costos asociados a las funciones logísticas y se definen como todos los costos que realiza una empresa para garantizar un buen nivel de servicio tanto para clientes, como para proveedores.

El proyecto, no afecta ni espera mejorar la cadena de abastecimiento en su totalidad, sin embargo, se pronostica una mejora en la eficiencia de

uno de los procesos que es de crítica importancia y junto con esto, reducción de costos en tareas que finalmente afectan a los costos logísticos totales. Cabe destacar que por las características del proyecto solo es posible evaluar la eficiencia operacional en términos de reducción de tiempos y maximización de volumen, puesto que no se cuenta con la información de los costos asociados al proceso.

3. Algoritmo Heurístico

Para saber qué es un algoritmo heurístico, primero es importante saber qué es la heurística como tal. El humano utiliza constantemente la heurística en su proceso de toma de decisiones reactivas, ya que al tener que reaccionar frente al problema y no tener tiempo para meditar la solución, prioriza la respuesta más rápida y conveniente, en vez de la respuesta más óptima, pues en caso de escoger esta última, este proceso tomaría mucho más tiempo. Cabe destacar que este proyecto se encuentra en el contexto investigativo de ciencias computacionales y matemáticas, en el libro *“Heuristics, History of”* se señala que aunque la definición de heurística varíe según el contexto de estudio, en el área relacionado con las ciencias computacionales y matemáticas se estudia la heurística como el método para encontrar una respuesta satisfactoria, dados posibles escenarios futuros y condiciones, como recursos escasos y/o soluciones informáticas inviables (Hertwig & Pachur, 2015). Basado en lo anterior, el proyecto se inclina por el desarrollo de un algoritmo heurístico, debido a que las empresas cuentan con tiempo y recursos limitados, por lo que encontrar un equilibrio entre la optimización de estos, y la calidad de los servicios y productos es una prioridad.

El objetivo principal de los algoritmos heurísticos es, según Sangaiah et al. (2020): “lograr una solución óptima sin buscar en todo el espacio de soluciones” (p.3), por lo que la respuesta que entrega es mucho más rápida, a pesar de no ser absolutamente óptima, por esto las soluciones que los utilizan han tomado gran relevancia en los últimos años, a pesar de que su implementación sea más complicada (Sangaiah et al., 2020).

4. Optimización

La optimización suele ser uno de los términos más utilizados dentro de cualquier tipo de organización, pero sobre todo en aquellas que están asociadas a distribución, ya que se busca encontrar el mejor método, gastando la menor cantidad de recursos. Por esto, las empresas de distribución buscan optimizar la mayor cantidad de procesos dentro de la cadena de abastecimiento, logrando así disminuir costos y ahorrar recursos.

Alimentos del Pedregal realiza la distribución mediante camiones que llevan pallets con cajas de productos, para llegar a cada sala de venta con sus respectivos pedidos. Esto conlleva tiempo y costos asociados a los recursos que se utilizan y los procesos realizados en cada operación logística. Es por esto, que optimizar incluso un único proceso logístico, genera impacto en toda la logística de distribución. En esta misma línea, Gajda, Trivella, Mansini, & Pisinger (2022), señalan que la optimización en procesos logísticos va tomando relevancia en los mercados de manera incremental, ya que ayudan a entregar servicios de calidad y realizar

distribuciones eficientemente, reduciendo así el uso de recursos e incluso la contaminación ambiental.

Para efectos de este proyecto, se quiere trabajar principalmente en la optimización de espacio, que se enfoca en maximizar la cantidad de cajas que se ubican sobre los pallets, y en la optimización del tiempo, que se enfoca en minimizar la demora del proceso que encuentra la maximización antes mencionada. Es por esto que con el algoritmo se busca armonizar ambas optimizaciones y entregar una buena respuesta en un tiempo adecuado, lo que generalmente se olvida en el problema de carga de contenedores (CLP en inglés; PCC en español) además, Gajda et al. (2022) señalan que en la mayoría de las soluciones de estos problemas se preocupan sólo de abarcar la optimización de espacios, dejando de lado necesidades y especificaciones de las organizaciones, como en este caso es la demora del proceso.

4.1. Pyomo

Para optimizar, tanto en la literatura académica como empresarial se mencionan y utilizan distintos paquetes de software y/o librerías para formular modelos de optimización, los cuales son modelos matemáticos que se utilizan en distintos ámbitos de estudio y que facilitado por un lenguaje de programación permite analizar diferentes situaciones bajo ciertas condiciones, el que debe tener una finalidad ya sea buscar el valor mínimo o el valor máximo, lo que depende de lo que se especifique en un principio como logro (Hart et al., 2017). Uno de los paquetes de software más utilizados para los modelos de optimización es Pyomo, un lenguaje de programación algebraica (AML) que está basado en Python y es de código abierto y según Hart et al. (2017), “El paquete Pyomo

incluye componentes de modelado que son necesarios para formular un problema de optimización: variables, objetivos y restricciones, así como otros componentes de modelado que comúnmente son compatibles con AML modernas, incluidos conjuntos de índices y parámetros” (p. 29).

Como se mencionó anteriormente, Pyomo es una herramienta potente para soluciones de optimización y tiene un amplio reconocimiento tanto en el ámbito académico como en el empresarial, además de estar basada en Python que según Hart et al. (2017), es un lenguaje de programación bien documentado y muy intuitivo, acompañado de que se puede utilizar para problemas de diversa complejidad y hay muchos aportes de terceros que pueden ser de gran ayuda.

5. Problema de carga de contenedores (CLP / PCC)

El problema de carga de contenedores según Gajda et al. (2022), es un problema de empaque de tres dimensiones de objetos, por lo general rectangulares, que se ubican dentro de un lugar de depósito o contenedor de mayor tamaño siguiendo restricciones previamente dadas. En la literatura se presenta como uno de los problemas más comunes de optimización y su solución apunta a maximizar el espacio que utiliza la carga dentro del contenedor, que en este caso sería el pallet. Algunos autores que han publicado artículos relacionados al Problema de carga de contenedores son Pisinger (2002), Lim, Ma, Qiu, & Zhu (2013) y Liu, Tan, Xu, & Liu (2014). Es importante destacar que se deben considerar: las tres dimensiones que tienen las cajas, los pallets o los camiones, es decir, largo,

ancho y volumen de cada uno de estos (Véase Figura 1); y la ortogonalidad de la carga, que quiere decir que cada cara del paquete es paralela a uno de los lados del contenedor (Véase Figura 2).

Figura 1 – Dimensiones de las cajas (Fuente Gajda et al, 2022) Traducción propia.

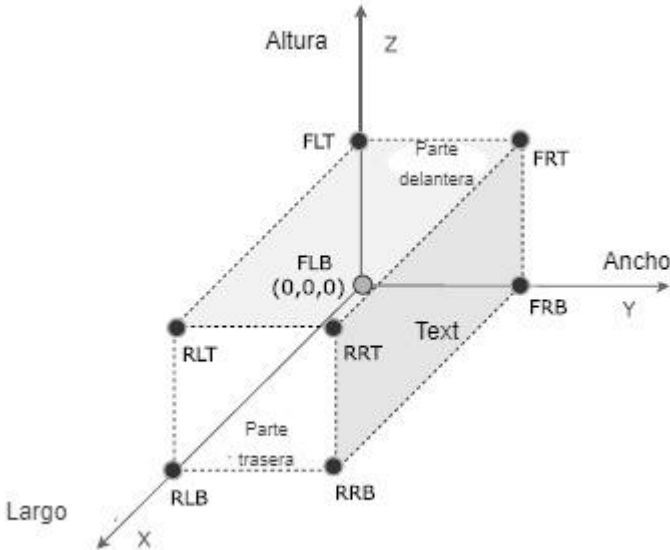
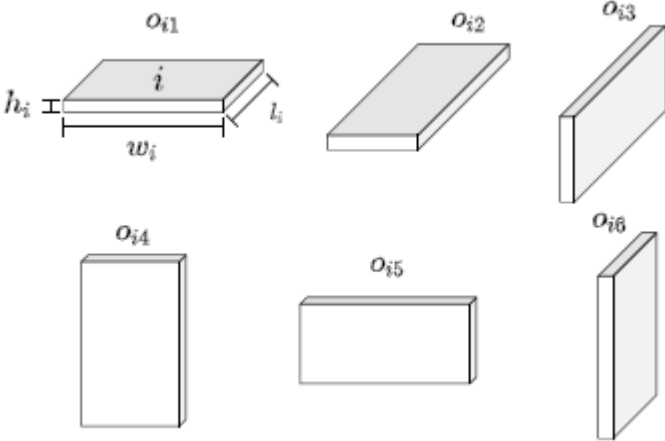


Figura 2 – Seis orientaciones de elementos ortogonales (Fuente Gajda et al., 2022)



Jugović (2020) menciona que maximizar la cantidad de unidades ubicadas en un contenedor más grande, es de alta complejidad y en todos los casos hay más de una respuesta al problema. Esto se explica porque si bien, puede que haya una solución que maximice completamente o más eficientemente el espacio, hay que considerar otros factores. Gehring & Bortfeldt (1997) señalan que para la organización de cajas en contenedor se deben considerar 5 restricciones además del volumen: orientación, colocación superior, peso, estabilidad y equilibrio. Estas restricciones se pueden alinear con las especificaciones del proyecto, porque no todo es completar el volumen máximo, si no tener en consideración distintas variantes. Para efectos del proyecto, se identifican las restricciones necesarias, pudiendo ser estas u otras, las cuales se especifican en los requerimientos del algoritmo.

Los algoritmos que entregan solución al problema se pueden clasificar principalmente en tres clases, las cuales son: método constructivo, el cuál va agregando cajas al contenedor, hasta no poder agregar más; el método de dividir y conquistar, que divide el contenedor en pequeños contenedores, da solución a cada uno de ellos y finalmente los une; y por último, un método de búsqueda local, que escoge una solución lista y aplica diferentes métodos para llegar a otras soluciones (Guerrero Fuentes & Núñez Parra, 2016).

Según las necesidades de este proyecto, se escoge una de las anteriores clases, para comenzar a desarrollar el algoritmo y se aborda cada una de las restricciones antes mencionadas u otras, para que la solución sea lo más personalizada posible, a la vez que rápida y óptima.

El problema de carga de contenedores está presente en casi todas las industrias, debido a que están en un constante crecimiento, por lo que cada vez toma más relevancia, y a pesar de tener mucha literatura con el

objetivo de dar solución a este problema, aún no es suficiente. Encontrar la solución óptima a este problema demanda mucho tiempo, por esto la heurística toma un rol protagonista, dando soluciones aceptables en tiempos aceptables.

Metodología

La metodología utilizada en este proyecto es la **metodología de desarrollo incremental**, dentro de este capítulo se describe y explica el desarrollo de esta. Se escogió esta metodología y no otra, debido a que se adapta de mejor manera a los objetivos de esta tesis. Según Sommerville (2011), es una buena metodología para el desarrollo de software en la cual se realizan las iteraciones que se consideren necesarias y sus respectivas correcciones de fallas hasta llegar a la versión final. Esta definición apunta a los objetivos presentados para este proyecto, ya que se pretende trabajar en varias versiones del algoritmo, de manera que constantemente se pueda presentar una mejora e ir agregando diferentes funcionalidades a este.

Primero se realiza la etapa de **bosquejo de descripción** en la cual se trabaja en una versión abstracta del producto, y se recolecta la información necesaria para comenzar a desarrollarlo. Luego, se desarrollaron las distintas fases y versiones del producto las que cuentan con sus propias actividades concurrentes, tales como **Especificación, Desarrollo y Validación** (Véase Figura 3).

Figura 3 – Iteraciones metodología de desarrollo incremental (Fuente Somerville., 2011)



1. Bosquejo de descripción

En esta fase se realizó un levantamiento de requerimientos de software para establecer los objetivos del sistema y las funciones que debe tener. Para esto se utilizó un formato facilitado por el profesor guía y se recolectó la siguiente información: requerimientos funcionales, casos de uso, alcances del producto, mockups y los recursos necesarios para el desarrollo del proyecto. La información fue recolectada en una reunión introductoria (1ra reunión) realizada de manera presencial, la que tuvo una duración de 1 hora aprox. En esta, participó el profesor guía, alumnos tesistas y la contraparte que está compuesta por Mario Lértora, gerente de operaciones, y Jorge Gallego, jefe de logística. Este documento fue presentado al profesor guía para su corrección y luego a la contraparte para su aprobación (Véase Anexo 1).

Se identificaron como requerimientos funcionales los siguientes hitos, los que deben ser cumplidos por el sistema a desarrollar:

- Recibir como input (entrada) el listado de productos a ordenar sobre el pallet.
- Generar una tarja, con el orden y ubicación de cada producto sobre el pallet.
- Tener las dimensiones de las cajas de cada producto previamente cargados para realizar la asignación.

Todo lo anterior, debe considerar las siguientes restricciones:

- Peso de los productos (pesados más abajo, livianos más arriba).
- Dimensiones de pallet y altura máxima del camión.
- Orientación de las cajas (algunas cajas se podrían ubicar en posición horizontal, más no vertical).

- Peso máximo del pallet.

Además, se generaron diversos casos de uso para tener más claridad de los diferentes procesos y lo que se necesita (Véase Figuras 4, 5 y 6).

Figura 4 – Caso de uso – Palletizar pedido (Autoría propia, 2022)

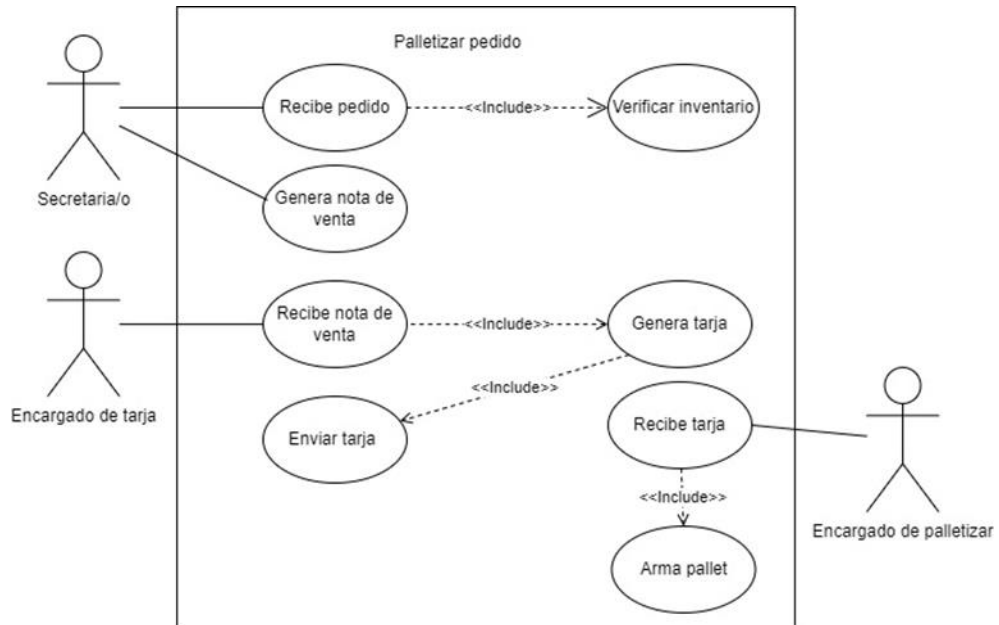


Figura 5 – Caso de uso – Generar tarja (Autoría propia, 2022)

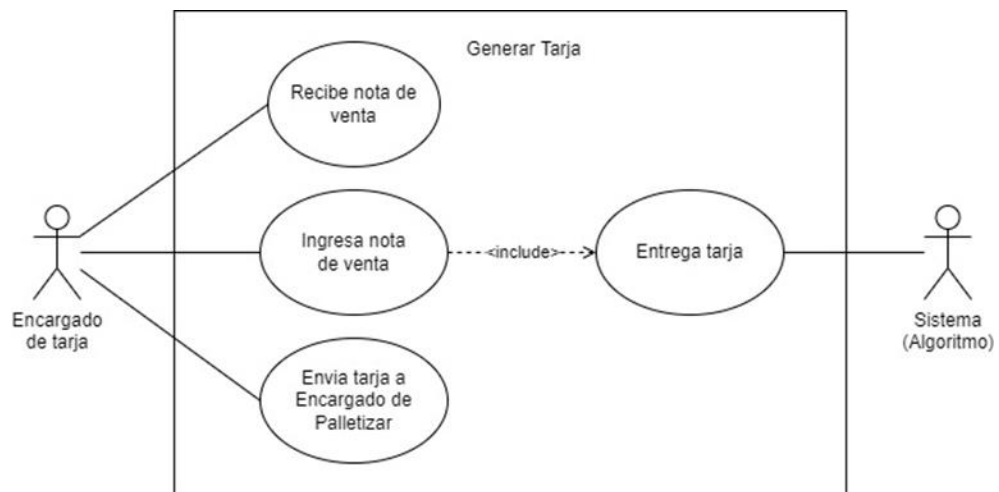
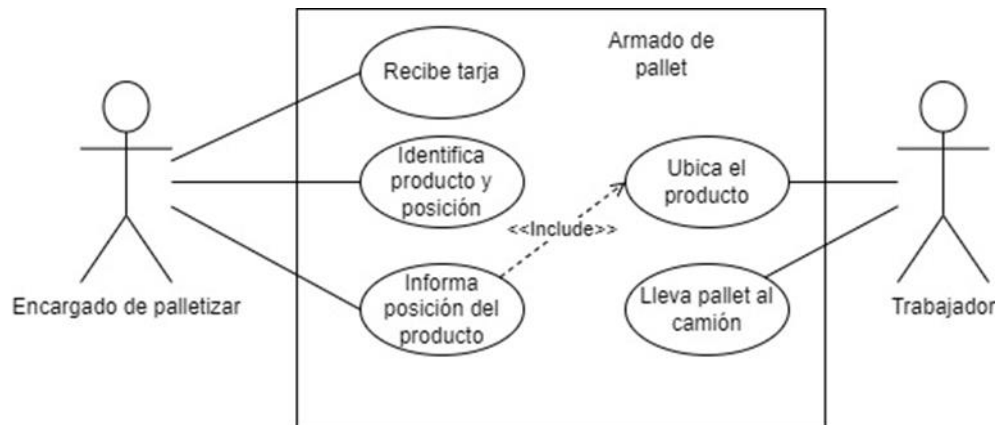


Figura 6 – Caso de uso – Armado de pallet (Autoría propia, 2022)



2. Planteamiento de la problemática

Para darle solución al problema, a través de la programación, se decide por un método de capas o niveles donde la simulación de ubicación de cajas sobre el pallet se realizará en distintos diagramas (tantos como niveles tenga) con formato 2D, este método se asocia a la estrategia de “dividir y conquistar”, la cual consiste en segmentar el contenedor, de tal manera que se acomoden las cajas en sectores más pequeños y saber lidiar con el espacio residual que dejan los diferentes segmentos.

3. Modelo Matemático

Para la modelación matemática del problema se definen parámetros, variables, función objetivo y restricciones, los cuales son obtenidos tanto por datos que entrega la empresa Alimentos del Pedregal, como por

necesidades que surgen para llegar a la solución propuesta. Cabe destacar que las restricciones no en todos los casos reflejan necesidades físicas del problema, como por ejemplo que las cajas se encuentren dentro del pallet, sino que también se requiere que el modelo ayude para que la programación del algoritmo cumpla requisitos reales, como por ejemplo que una caja no atraviese otra. Los detalles de la problemática y el modelo matemático fueron discutidos y resueltos en una reunión intermedia (2da reunión) de una duración aproximada de 1 hora y 30 minutos. En esta, participó el profesor guía, alumnos tesistas y la contraparte que está compuesta por Mario Lértora, gerente de operaciones, Jorge Gallego, jefe de logística, y esta vez se sumó Martín Cárcamo, director de Innovación.

4. Algoritmo Heurístico

Para el desarrollo del algoritmo, fue fundamental el levantamiento de requerimientos, pues en este se definieron las funciones básicas que debía tener el algoritmo, como la lectura de datos tanto del listado de productos, como de la nota de venta; además se marcaron los límites del proyecto y cuánto debía abarcar.

En el desarrollo del algoritmo fue en dónde más siguió al pie de la letra la metodología de desarrollo incremental, ya que en cada versión de este se realizaba una especificación, la cual planteaba qué requerimientos debía cubrir esa versión y qué funciones debía tener, luego se entraba a una fase de desarrollo, en la cual se desarrollaba el código, con el fin de que este logrará lo previamente planteado en la fase de especificación y

finalmente, se entró en una fase de validación, la cual se enfocó en encontrar las fallas del programa y ver si efectivamente se cumplieron los requerimientos comprometidos a esa versión. En caso de que estos no sean cumplidos, se vuelve a la fase de desarrollo, hasta perfeccionar la versión, de manera que, al continuar con las siguientes versiones, que tienen otros objetivos y especificaciones, no se arrastren problemas previos.

La versión final del algoritmo se presentó a la contraparte en la reunión de muestra de resultados (3ra reunión), la cual tuvo una duración aproximada de 2 horas. En esta, participó el profesor guía, alumnos tesistas y la contraparte que está compuesta por Mario Lértora, gerente de operaciones, Jorge Gallego, jefe de logística, y Martín Cárcamo, director de Innovación. Los alumnos tesistas explicaron las funcionalidades y recomendaciones para el uso del algoritmo a la contraparte, mostrando los outputs propuestos y la solución final, siempre abiertos a mejoras. Al finalizar la reunión, la contraparte se mostró conforme, y realiza correcciones y recomendaciones menores que se solucionaron de forma inmediata.

Además de la metodología utilizada para el desarrollo del algoritmo, se utilizó otra metodología para realizar la evaluación de eficiencia operacional, la cual será descrita a continuación:

5. Metodología de la evaluación de eficiencia operacional y estimación de costos asociados

Para Alimentos del Pedregal, la necesidad del desarrollo del sistema es para disminuir costos que están relacionados al tiempo de proceso de armado de los pallets. Es por esto, que se propone comparar los tiempos del sistema actual, un software comercial que se intentó implementar y la heurística desarrollada en el proyecto, relacionados a la ubicación conceptual de las cajas sobre los pallets y hacer un estimado del impacto que tendría en las horas hombre y el costo asociado a estas.

Los tiempos del sistema actual y el software comercial que se intentó implementar fueron informados por la contraparte. Por otro lado, el algoritmo desarrollado armó un pallet conceptualmente, con un conjunto de cajas provenientes de una nota de ventas real que se utilizó en la empresa y se evaluó el tiempo que demoró en entregar una solución. Además, el algoritmo desarrollado realiza iteraciones parametrizables de la ubicación de las cajas y se queda con la que más volumen del pallet utiliza.

Por otro lado, también se realizaron mediciones respecto al volumen utilizado en los pallets que fueron armados por las tarjas obtenidas del sistema actual y comparado con el volumen utilizado de las tarjas arrojadas por la heurística desarrollada.

Presentación y análisis de los resultados

1. *Desarrollo del modelo matemático*

Realizada la especificación, junto al desarrollo del algoritmo de optimización se debe crear un modelo matemático que represente el problema real y que permita solucionarlo con la programación. El modelo matemático desarrollado es el siguiente:

Modelo matemático

Para llevar a cabo el proyecto, es importante realizar la representación matemática del problema previo a la programación de este. Las dimensiones tanto del pallet (P) como las cajas (i) del conjunto de éstas (B), pertenecen a los reales positivos cúbicos, es decir:

- $P(L, A, H) \in R_+^3$
- $i \in B(l_i, a_i, h_i) \in R_+^3$

donde, L , A y H en su correspondiente orden son el largo, ancho y altura del pallet; y, l , a y h son largo, ancho y altura de las cajas.

1.1. Parámetros

Dicho lo anterior, tenemos que cada caja contará con sus respectivos parámetros, que son datos que nos proporciona la empresa, los cuales son:

- $l(B_i)$: largo de la caja cuando fue medida por Alimentos del pedregal.
- $a(B_i)$: ancho de la caja cuando fue medida por Alimentos del pedregal.
- $h(B_i)$: altura de la caja cuando fue medida por Alimentos del pedregal.
- $v(B_i)$: volumen de las cajas que se obtiene multiplicando todas las dimensiones de estas.

- $w(B_i)$: peso de las cajas proporcionado por Alimentos del pedregal.
- $V(P)$: volumen del pallet que se obtiene multiplicando todas las dimensiones de este.
- $W(P)$: peso máximo permitido para cargar el pallet, proporcionado por Alimentos del pedregal.

1.2. Variables

Si bien, cada caja tiene sus dimensiones, al ubicarse en el pallet puede que tenga una orientación distinta a cuando fue realizada la medición en un principio. Es por esto que se hace necesario considerar:

- l_i : largo de la caja ubicada en el pallet. (Eje x).
- a_i : ancho de la caja ubicada en el pallet. (Eje y).
- h_i : alto de la caja ubicada en el pallet. (Eje z)
- vx_i : posición del vértice de la caja i más cercano al valor 0 en el eje x.
- vy_i : posición del vértice de la caja i más cercano al valor 0 en el eje y.
- p : niveles del pallet que se crean con relación a la caja i de menor altura del nivel.
- $v(p)$: volumen del nivel p que se crea con la suma de los volúmenes de las cajas i .
- $pp(B_i)$: puntaje de prioridad que se crea con la suma de peso y dimensiones normalizados de las cajas i .

1.3. Función objetivo

La representación en lenguaje matemático de lo que buscamos como objetivo con el modelo es:

$$Max \sum_p^n \sum_i^n v(B_{p,i}) * m_i$$

donde se entiende que se debe maximizar la sumatoria doble del volumen de las cajas i , del nivel p , del conjunto B y la variable m es un valor binario (0 no está, 1 si está) que indica que la caja se ubicó en el pallet.

1.4. Restricciones

Restricción 1: Conjunto de cajas (B_i) posee un volumen que pueda estar completamente dentro del pallet (P):

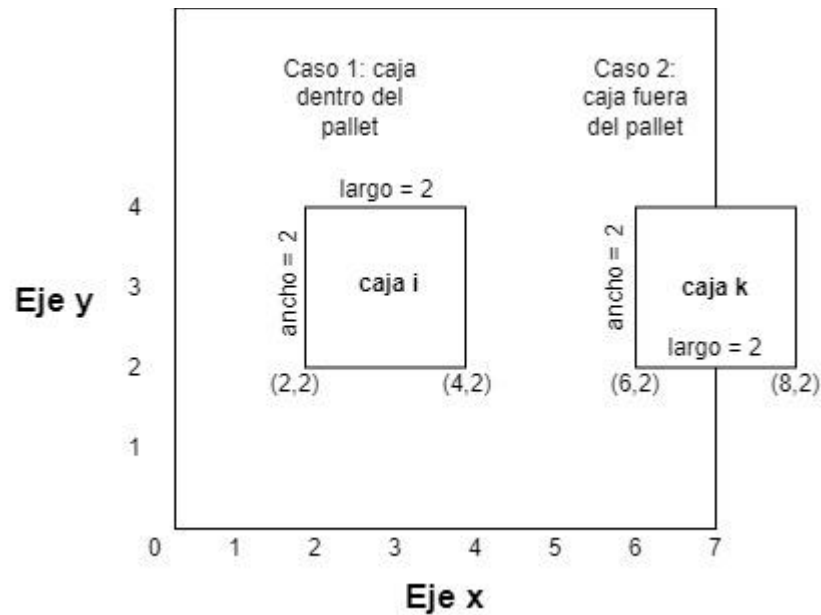
$$v(B_i) \leq V(P)$$

Restricción 2: Las cajas se deben ubicar dentro de las dimensiones del pallet; una vez ubicada la caja dentro del pallet, es decir, si m_i verdadero, aplicamos las siguientes condiciones:

$$vx_i \geq 0, vx_i + l_i \leq L$$

$$vy_i \geq 0, vy_i + a_i \leq A$$

Figura 7– Diagrama de restricción de cajas dentro del pallet (Autoría propia, 2022)



La Figura 7 muestra dos casos:

El caso 1 cumple la restricción de estar dentro del pallet visualmente, y si verificamos con la fórmula:

$$2 \geq 0, 2 + 2 \leq 7$$

$$2 \geq 0, 2 + 2 \leq 4$$

Ambas condiciones son verdaderas, por lo que la caja *i* está dentro de los límites del pallet.

Por otro lado, el caso 2 no está dentro, por lo que verificamos con la fórmula:

$$6 \geq 0, 6 + 2 \leq 7$$

$$2 \geq 0, 2 + 2 \leq 4$$

Por lo que la primera condición confirma que la caja *k* se encuentra fuera de los límites establecidos

Restricción 3: El peso total de las cajas no puede superar el máximo permitido por el pallet.

$$w(B_i) \leq W(P)$$

Restricción 4: Las cajas no pueden ser apiladas sobre otras que poseen un peso inferior.

$$pp(B_i) \geq pp(B_{i+1})$$

Restricción 5: Las cajas no se pueden ubicar en todas las orientaciones físicamente posibles. Las orientaciones que pueden tomar las cajas, una vez ubicadas en el pallet, se representan así:

$$(l_i, a_i, h_i) = (l(B_i), a(B_i), h(B_i)) \forall$$

$$(l_i, a_i, h_i) = (a(B_i), l(B_i), h(B_i)) \forall i \in B$$

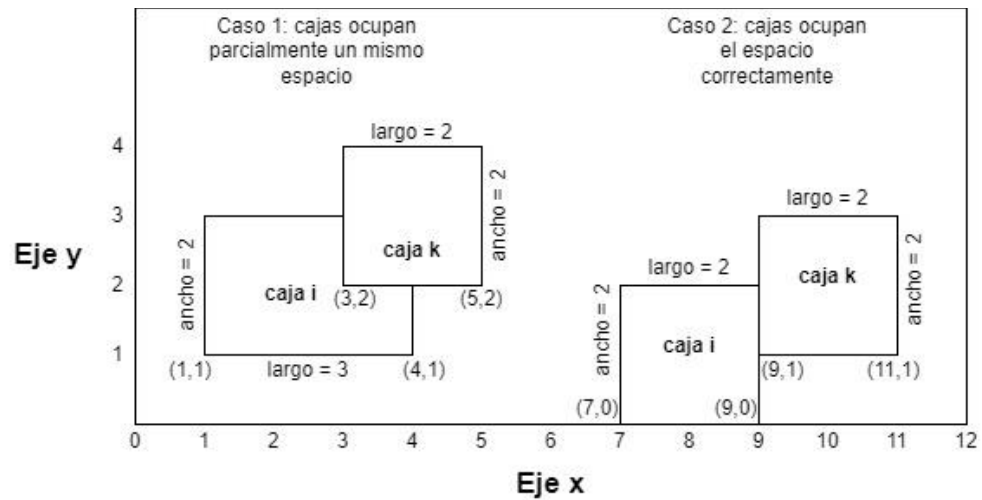
Restricción 6: Las cajas no pueden utilizar el mismo volumen, es decir no se pueden traslapar. (En el mundo real se puede obviar, puesto que una caja no puede o no debería atravesar físicamente otra, pero es necesario dejarlo explícito en el modelo matemático para la posterior programación).

Para dos cajas (i, k) que se ubican en el pallet, es decir $m_i = m_k$, podemos representar matemáticamente esta restricción de la siguiente forma:

$$(vx_i + l_i \leq vx_k) \vee (vy_i + a_i \leq vy_k) \forall k, i \neq k$$

Para que haya traslape, se debe seguir la lógica proposicional de la fórmula anterior. En la Figura 8 se explica de mejor forma:

Figura 8– Diagrama de restricción de traslape (Autoría propia, 2022)



Como se puede ver en el caso 1, hay traslape. Siguiendo la fórmula anterior, la superposición de cajas se explica porque el vértice de la caja i en el eje x (vx_i) más cercano al origen es = 1 y si se le suma el largo (l_i) = 3 y se verifica si se cumple la condición de que $vx_i + l_i \leq vx_k$, cuyo valor es = 3, entonces:

$$1 + 3 \leq 3 \rightarrow F$$

Además, se verifica también que $vy_i + a_i \leq vy_k$, cuyos valores entregan el resultado:

$$1 + 2 \leq 2 \rightarrow F$$

Por lo tanto, al ser ambas falsas, se aplica la restricción.

Por otro lado, tenemos el caso 2, donde no existe traslape. Siguiendo la proposición lógica de la fórmula, se representa numéricamente:

$$(7 + 2 \leq 9) \vee (0 + 2 \leq 1) \rightarrow V$$

Cabe destacar que, al poseer disyunción lógica como símbolo intermedio, basta que se cumpla una condición para que la restricción no aplique.

Las dimensiones de las cajas están definidas según la información obtenida de la lista de productos y las de los pallets son las siguientes:

$$100 \text{ cm} * 120 \text{ cm} * [180 \text{ cm} , 220 \text{ cm}]$$

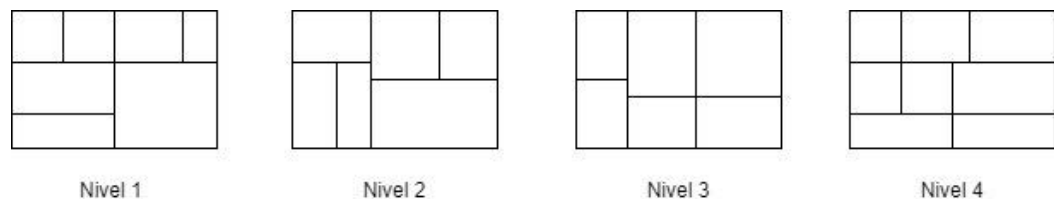
Donde la altura mínima que se puede apilar es de 180 cm y la máxima es de 220 cm.

2. Desarrollo de las diferentes versiones del programa.

2.1. Simulación de la ubicación de cajas por nivel sobre el pallet.

La siguiente figura ejemplifica la simulación de ubicación de cajas por nivel sobre un pallet:

Figura 9 – Diagrama de posicionamiento de cajas sobre pallet (Autoría propia, 2022)



La Figura 9, simula un contenedor con 4 niveles de cajas apiladas, en donde la idea es ubicar cada caja, la cual tiene su respectivo código asociado, sobre el pallet, según el diagrama previamente presentado. En este caso ya estarían consideradas las restricciones respectivas, como el peso que aguanta cada caja o las diferentes dimensiones.

2.2. Primera versión

La versión inicial del programa se encontraba enfocada a la carga del archivo que contiene el listado de productos y el manejo simple de estos datos dentro del algoritmo. Esto es importante, debido a que es necesario tener en cuenta las dimensiones de las cajas de productos y del pallet que se utilizarán, para saber cómo acomodarlos posteriormente. Es por esto que se generó la codificación adecuada para la obtención de los datos relevantes de las hojas Excel que fueron proporcionadas por Alimentos del Pedregal, para esto se tenían dos opciones de librerías, las cuales eran xlrd y pandas. Finalmente se optó por utilizar pandas, puesto que es más compatible con las necesidades del proyecto, además tiene una mayor variedad de funciones y soporta diferentes tipos de archivos Excel.

En iteraciones secundarias de la misma versión, se concentró más en la organización del código, con el fin de facilitar su entendimiento para personas externas al proyecto y también el manejo del código en un futuro. Además, se agregó una función para el manejo de las cajas dentro del contenedor, pero sin considerar ninguna de las restricciones especificadas previamente. Importante destacar que esta “carga” de cajas al contenedor se realizó según una lista de productos obtenida de una nota de ventas entregada por Alimentos del Pedregal.

```
#LIBRERIAS
import pandas

#CONFIGURACIÓN PANDAS
pd.options.mode.chained_assignment = None # default='warn'

#MAIN

df_datos = pandas.read_excel("lista dimensiones.xlsx") #objeto de tipo
dataframe
df_notadeventa = pandas.read_excel("nota de venta.xlsx")
```

Esta es la manera de cargar los Excel que contienen los datos asociados a los productos y almacenados en variables de tipo dataframe.

2.3. Segunda versión

La segunda versión del código se encuentra enfocada en darle una solución orientada a objetos al problema, por lo que tanto la caja como el pallet serán tratados como objetos dentro del código. La caja tendrá atributos como su id, su nombre, sus medidas, su peso y su volumen, por otro lado el pallet estará en constante modificación, tendrá atributos estáticos como sus medidas, el volumen máximo, la altura máxima y el peso máximo, pero también tendrá atributos variables, como lo sería una lista bidimensional que representaría la superficie del pallet como un plano cartesiano en donde se acomodarían las cajas, otro atributo variable sería una lista que contiene las cajas que van siendo ubicadas en el plano cartesiano y por último atributos como el peso y el volumen utilizado hasta el momento.

A diferencia de la primera versión, para facilitar el manejo de los datos, las variables de tipo dataframe se procesan para obtener listas con objetos de tipo caja, de la siguiente manera:

```

#LIBRERIAS
import pandas as pd

#CONFIGURACIÓN PANDAS
pd.options.mode.chained_assignment = None # default='warn'

#listas

cajastodas=[]
listnotadeventa=[]

#Exportar documento
df_datos= pd.read_excel("C:/Users/ani_1/Desktop/proyecto tesis/proyecto
definitivo/lista dimensiones.xlsx")
df_notadeventa = pd.read_excel("C:/Users/ani_1/Desktop/proyecto
tesis/proyecto definitivo/nota de venta.xlsx")

#MAIN

#se llena el array con cajas
for i in range(len(df_datos)):
    c_ins=df_datos.iloc[i]
    cajastodas.append(caja.caja(c_ins.largo,c_ins.alto,c_ins.ancho,c_in
s.peso_bruto,c_ins.nombre))

```

La primera lista en llenarse es en la que se almacenan todas las cajas con sus respectivos datos.

```

for i in range(len(df_notadeventa)):
    c_ins=df_notadeventa.iloc[i]
    for cjs in cajastodas:
        if (cjs.nombre==c_ins.producto):
            for x in range(c_ins.cantidad):
                cjas=caja.caja(cjs.largo,cjs.alto,cjs.ancho,cjs.peso,cj
s.nombre)
                listnotadeventa.append(cjas)

```


La segunda lista en llenarse corresponde a la nota de venta, en la cual se agregan las cajas según la cantidad y el tipo que se encuentra especificado en el Excel.

En esta versión se comenzaron a agregar las cajas al plano cartesiano bajo la siguiente lógica:

- Se recorre la lista y se intentan agregar las cajas al plano cartesiano. Una vez ubicada la caja, se borra de la lista.
- Se evalúan tres condiciones, que la caja no esté ubicada aún, que no exceda el volumen máximo y que no exceda el peso máximo.
- Se recorre el plano cartesiano desde la coordenada X y la coordenada Y dejado por la caja anterior (en caso de ser la primera caja, estas toman el valor de 0,0), hasta las nuevas X, Y obtenidas al agregar esta caja.
- Cuando se “pueda” ubicar la caja se modifica el plano cartesiano con el id de la caja agregada
- Finalmente se modifica la coordenada X, según el largo de la caja agregada y en el caso de que ya no se pueda agregar la caja hacía el eje x, este pasa a ser 0 y se modifica la coordenada Y según el ancho de la primera caja agregada (Véase Figura 10).

Figura 10– Ejemplificación de cajas posicionadas en plano cartesiano (Autoría propia, 2022)

CAJA 1	CAJA 2			CAJA 3	CAJA 4	
CAJA 5	CAJA 6					
				CAJA 7		

Esta lógica tenía grandes falencias, ya que se generan espacios residuales que no están siendo aprovechados (zona gris) o se produce un traslape, como es en el caso de la caja 3 y la caja 7 que se sobreponen (zona roja).

2.4. Tercera versión

Esta versión se enfocó en la optimización de la acomodación de las cajas en el espacio disponible dentro del pallet, se divide en dos grandes áreas, la aplicación del puntaje de prioridad a la base de datos y la implementación de Pyomo en el algoritmo.

Respecto al puntaje de prioridad, se crearon por la necesidad de mantener la estabilidad en los distintos niveles y para evitar que las cajas sean ubicadas arriba de otras de menor peso. Primero, se normalizaron los datos de las cajas (peso, largo, ancho, alto), y luego, se le otorgó un 70% de importancia al peso y un 30% a las dimensiones, puesto que el principal objetivo es que las cajas pesadas se ubiquen primero y el segundo objetivo es que las dimensiones sean parecidas en el mismo nivel. Cabe destacar que, se sumó un valor suficiente a todos los puntajes de prioridad para que el total no fuera un valor negativo, puesto que generaba errores en el código.

Respecto a Pyomo, en primera instancia se trabajó independiente del código, solamente se creó el modelo matemático con las restricciones que no fueron abordadas en la codificación y con la función objetivo antes declarada. Luego de realizar las primeras pruebas exitosas se implementó en la codificación del proyecto en su totalidad.

La lógica a seguir es la siguiente: Primero se le entrega al modelo realizado en Pyomo la lista con las cajas de la nota de venta y esta entrega una lista

con las cajas que el modelo considero que eran las más optimas de agregar al pallet, luego estas se intentan agregar al pallet y cuando ya no cabe ninguna caja más en el nivel, se vuelve a entregar la lista al modelo, pero ahora sin las cajas que ya están dentro del pallet. En relación al funcionamiento del puntaje de prioridad, en cada iteración de optimización, se ordena la lista entregada por el modelo según su puntaje, de manera que se les dé prioridad a las cajas según el puntaje más alto.

2.5. Versión Final

Esta versión está enfocada en perfeccionar cada una de las áreas abordadas en las versiones anteriores. Finalmente la codificación quedó en cuatro clases, estas son *caja.py*, en la cual está definida la caja como objeto, con sus atributos y funciones; *func.py*, en la cual se crearon variadas funciones según las necesidades del proyecto; *matrices.py*, es en la cual se define el pallet como objeto, con sus atributos y funciones, esta es una de las clases más densas, ya que aquí se realiza y almacena la acomodación de las cajas; y por último, *main.py*, que es la clase principal, desde donde se comienza a ejecutar el código.

La solución está entregada de la siguiente manera:

- Para facilitarle a la contraparte la carga de los archivos, esta se realiza con la librería Tkinter, por medio de un *filedialog*, por lo que quien ejecute el programa deberá seleccionar tanto el archivo de la base de datos, como la de la nota de venta.

```

archivoNdV=filedialog.askopenfilename(title="Abrir Nota de Venta",
initialdir="C:/",filetypes=(("Hojas de cálculo",
    "*.xlsx"),("Todos los Archivos", "*.*")))
archivoBBDD=filedialog.askopenfilename(title="Abrir BBDD",
initialdir="C:/",filetypes=(("Hojas de cálculo",
    "*.xlsx"),("Todos los Archivos", "*.*")))

```

- Comienzan a realizarse las iteraciones de optimización, las que están dadas por una variable al inicio del código, que para efectos de la entrega del código a la contraparte se definió en cinco. Cada iteración tiene un porcentaje de utilización del pallet, aquella que tenga el mayor porcentaje de las cinco iteraciones es la que será entregada en el output.
- Cada iteración de optimización termina de ejecutarse cuando ya están todas las cajas posicionadas en el pallet, cuando las cajas apiladas llegan a la altura máxima o cuando se llega al volumen máximo.
- Se ejecuta el modelo de optimización, obteniendo una nueva lista, la cual se ordena según su puntaje de prioridad, utilizando una función burbuja, esta lista ordenada será la que se comenzará a agregar al plano cartesiano.

```

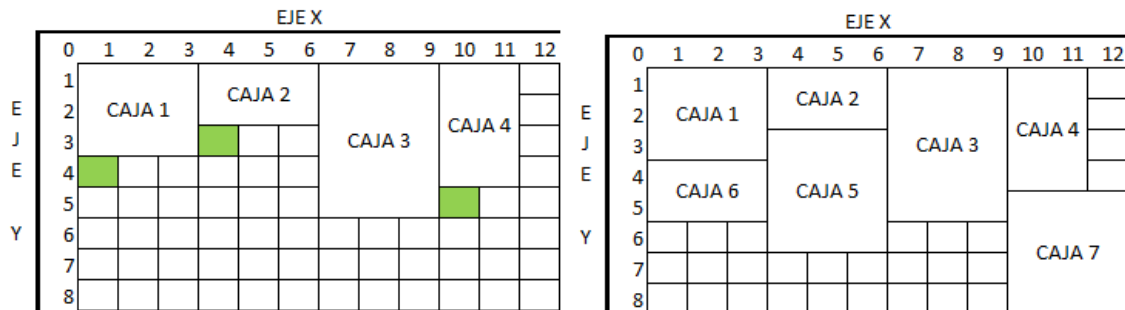
def burbuja(list):
    n = len(list)
    for i in range(n-1):
        for j in range(n-1-i):
            if list[j].ptjprioridad < list[j+1].ptjprioridad:
                list[j+1], list[j] =list[j], list[j+1]
    return list

```

Como fue mencionado en la segunda versión, la manera de acomodar las cajas no era la más óptima, por lo que ahora se generó otra lógica en donde se aprovechan los espacios residuales y se evita el traslape.

- En un principio se decide de manera aleatoria en que orientación se ingresará la caja al pallet, en caso de que no caiga, se prueba con la otra orientación. Si la caja no cae en ninguna de las dos orientaciones, se pasa a la siguiente, e itera hasta que alguna de las cajas entre al pallet. Cuando esto sucede, se vuelve al inicio de la lista para darle prioridad a las primeras cajas de la lista que no estén ubicadas en el pallet.
- Para acomodar las cajas se busca el punto vacío más cercano a las coordenadas 0,0, como se muestra en la Figura 11:

Figura 11- Ejemplificación de cajas posicionadas en plano cartesiano con nuevo método (Autoría propia, 2022)



- En el caso de la caja 5, el punto más cercano al origen en donde se puede ubicar la caja es (3,2), para la caja 6 es (0,4) y para la caja 7 es (9,5), por lo tanto, el algoritmo las ubica en esa posición.

Con este método también quedan espacios residuales, pero suelen quedar en los extremos de la superficie, por otro lado, para evitar el traslape, el programa se asegura de que la caja cae en el espacio antes de acomodarla, de lo contrario busca otro punto vacío en donde si caiga.

Para obtener el output final se utilizó la optimización desarrollada en la tercera versión, en un principio el output entregado por el programa eran dos archivos .txt, uno con las cajas ubicadas en el plano cartesiano y otro que contenía le leyenda de las cajas, pero luego de presentarle esta

propuesta a la contraparte se acordó entregar un único archivo .txt en donde se detallaba en cada nivel el tipo de producto y la cantidad de cajas de este que se ubicarían en el pallet.

Esta es la versión que fue entregada a la contraparte, luego de solucionar mínimos detalles que fueron conversados y acordados en la última reunión.

3. *Desarrollo de la evaluación de eficiencia operacional*

Las pruebas realizadas con el algoritmo desarrollado con notas de ventas reales muestran un desempeño favorable en tiempo. El Cuadro 1 muestra la comparación de tiempo en minutos de la carga de datos y del armado conceptual de una nota de venta:

Cuadro 1 – Tabla comparativa de tiempos (Autoría propia, 2022)

	Comparación de tiempos promedios en minutos	
	Carga de datos	Armado conceptual de la nota de venta
Sistema actual	-	15
Software comercial	5	1,5
Heurística desarrollada	1	1,3 (1 iteración) 5,7 (5 iteraciones)

Los resultados demuestran que la heurística desarrollada se demora menos en entregar una solución, por lo que se cumple el objetivo de entregar una solución óptima, pero no perfecta, en un tiempo aceptable. Además, la contraparte informó que, si bien el software comercial tenía un tiempo aceptable, la solución que entregaba en ocasiones era inviable, porque

ubicaba cajas que debían ir obligatoriamente horizontal en orientación vertical. Debido a las características del proyecto y su compleja implementación, no se realizaron mediciones del armado del pallet físicamente, ya que requiere de una adaptación de parte de todos los actores a esta nueva manera de realizar el proceso.

Por otro lado, respecto al volumen, se realizó una tabla comparativa (Véase Cuadro 2) con una nota de venta que fue dividida en tres tarjas, lo que quiere decir que se realizaron tres pallets, y se comparó con el porcentaje de ocupación de los pallets arrojados por la nota de venta ingresada al sistema.

Cuadro 2 - Tabla comparativa de volumen (Autoría propia, 2022)

	Volumen utilizado del pallet (%)		
	Pallet 1	Pallet 2	Pallet 3
Sistema actual	47%	65%	53%
Heurística desarrollada	67%	74%	25%

La tabla comparativa arroja el evidente aumento del uso del espacio disponible en el pallet, en comparación al espacio que se utiliza con el sistema actual, a pesar de que en Pallet 3 el uso es menor, esto se debe a que la cantidad de cajas disponibles para acomodar también es menor, por lo que se consiguió el objetivo de aumentar el volumen utilizado dentro del pallet.

3.1. Estimación de costos asociados

La contraparte informó que los trabajadores cumplen jornadas de 8 horas, de las cuales 6 horas y media son realmente productivas. También, mencionan que 4 trabajadores (1 supervisor y 3 armadores) arman

físicamente 3 pallets cada 45 minutos, donde se incluyen los 15 minutos del armado conceptual de las notas de venta. Es decir, se estiman aproximadamente 26 pallets armados por día trabajado con el sistema actual, con el supuesto que en promedio por cada nota de venta se arman 3 pallets. Y, por otro lado, con la heurística desarrollada se estima el armado de 32 pallets por día trabajado, ya que los 15 minutos del armado conceptual se reducirían a 6,7 minutos. Esto en la práctica puede tener variaciones, ya que solo son estimaciones, pero el mayor impacto, respecto a costos, se vería reflejado en la productividad de las horas hombre, ya que se redujo en un 18,4% el tiempo del armado de una nota de venta y 3 pallets, por lo que se puede decir que el armado de cada nota de venta tendrá un coste menor.

Conclusiones

Como resultado principal, se desarrolló un algoritmo heurístico para optimizar la ubicación de cajas sobre pallet para la empresa Alimentos del Pedregal. El algoritmo, entrega un output que permite a los trabajadores del área de armado de pallets ubicar las cajas de manera óptima, con relación a la estabilidad y capacidad de carga del pallet. La evaluación de la eficiencia operacional demostró que la ubicación conceptual de las cajas con la heurística desarrollada se realiza más rápido que con el sistema actual y en un tiempo similar al software comercial que se intentó implementar. Y también, el algoritmo desarrollado permite maximizar el volumen del pallet en comparación al sistema actual.

Además, se destaca la colaboración de la contraparte al facilitar la información necesaria para llevar el proyecto a cabo, puesto que contaban con una base de datos que contenía las dimensiones de todas las cajas necesarias (productos congelados). Sin embargo, se hizo necesario depurar la base de datos en detalles menores o de poca relevancia, pero que retrasaban la carga de esta al tener que hacer una búsqueda más exhaustiva en los datos.

Por otro lado, se recomienda a la empresa que las notas de ventas (Input) comiencen a tener los ID de los productos, ya que la búsqueda de estos en la base de datos, para obtener sus dimensiones, podría agilizarse a nivel de CPU.

En relación con la problemática, la gran variedad de trabajos que entregan solución al problema de carga de pallets y/o contenedores tienen distintos enfoques. Como alumnos tesistas, nos gustaría destacar que nuestro enfoque fue en todo momento entregar una respuesta rápida y aceptable, la esencia de la heurística.

Los aprendizajes obtenidos con el proyecto superaron nuestras expectativas y aunque el resultado final es perfeccionable, como considerar y disminuir los movimientos de trabajadores hacia la bodega de inventario, estamos conformes porque significa un gran aporte para la optimización de la carga de cajas sobre pallets.

Nuestros intereses para futuros trabajos de este problema están enfocados en que el output sea más acercado a la realidad, es decir, modelado 3D. Sin embargo, se puede perder eficiencia y rapidez de respuesta, por lo que se recomienda encontrar una armonía entre el modelado gráfico y la eficiencia de la palletización.

Bibliografía

- Alonso, M. T., Alvarez-Valdes, R., Iori, M., Parreño, F., & Tamarit, J. M. (2017). Mathematical models for multicontainer loading problems. *Omega*, *66*, 106–117. <https://doi.org/10.1016/j.omega.2016.02.002>
- Andrejic, M., Milenkovic, M., & Sokolovic, V. (2010). Logistics information system. *Vojnotehnicki Glasnik*, *58*(1), 33–61. <https://doi.org/10.5937/vojtehg1001033A>
- Baldi, M. M., Perboli, G., & Tadei, R. (2012). The three-dimensional knapsack problem with balancing constraints. *Applied Mathematics and Computation*, *218*(19), 9802–9818. <https://doi.org/10.1016/j.amc.2012.03.052>
- Ballou, R. (2004). *Logística* (5ª ed.). México: Pearson Educación.
- Carro Paz, R., y González Gómez, D. (2013). *Logística Empresarial*. Recuperado a partir de <http://nulan.mdp.edu.ar/id/eprint/1831>
- Cordero Balind, V., Torres Argüelles, V., Hernández Gómez, A., & Ibarra Mejía, G. (2015). Sistema de información en el proceso de logística inversa, revisión de literatura. *Cultura Científica y Tecnológica*, *55*(1), 46–59. Recuperado a partir de <https://erevistas.uacj.mx/ojs/index.php/culcyt/article/view/745>
- Dell'Amico, M., & Magnani, M. (2021). Solving a Real-Life Distributor's Pallet Loading Problem. *Mathematical and Computational Applications*, *26*(3), 53. <https://doi.org/10.3390/mca26030053>
- Gajda, M., Trivella, A., Mansini, R., & Pisinger, D. (2022). An optimization approach for a complex real-life container loading problem. *Omega*, *107*, 102559. <https://doi.org/10.1016/j.omega.2021.102559>

- García-Cáceres, R., Vega-Meja, C., & Caballero-Villalobos, J. (2011). Integral Optimization of the Container Loading Problem. En *Stochastic Optimization - Seeing the Optimal for the Uncertain*. InTech. <https://doi.org/10.5772/15744>
- Gehring, H., & Bortfeldt, A. (1997). A Genetic Algorithm for Solving the Container Loading Problem. *International Transactions in Operational Research*, 4(5–6), 401–418. <https://doi.org/10.1111/j.1475-3995.1997.tb00095.x>
- Guerrero Fuentes, K., & Núñez Parra, E. (2016). *Problema de Carga de Contenedores (Container Loading Problem)*.
- Gzara, F., Elhedhli, S., & Yildiz, B. C. (2020). The Pallet Loading Problem: Three-dimensional bin packing with practical constraints. *European Journal of Operational Research*, 287(3), 1062–1074. <https://doi.org/10.1016/j.ejor.2020.04.053>
- Hart, W. E., Laird, C. D., Watson, J.-P., Woodruff, D. L., Hackebeil, G. A., Nicholson, B. L., & Sirola, J. D. (2017). *Pyomo — Optimization Modeling in Python* (Vol. 67). Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-58821-6>
- Hertwig, R., & Pachur, T. (2015). Heuristics, History of. En *International Encyclopedia of the Social & Behavioral Sciences* (pp. 829–835). Elsevier. <https://doi.org/10.1016/B978-0-08-097086-8.03221-9>
- Jugović, A. (2020). Economic Impact of Container-loading Problem. *Transactions on Maritime Science*, 9(2). <https://doi.org/10.7225/toms.v09.n02.010>
- Leitão, F., Silva, W., Silva, R., & Brisola, M. (2020). Levantamento dos custos logísticos da distribuição física da polpa de frutas em uma cooperativa de agricultores familiares. *Custos E @Gronegócio On Line*, 16(1808–2882),

397–420. Recuperado a partir de <http://www.custoseagronegocioonline.com.br/especialv16/OK%2017%20polpa.pdf>

Lim, A., Ma, H., Qiu, C., & Zhu, W. (2013). The single container loading problem with axle weight constraints. *International Journal of Production Economics*, *144*(1), 358–369. <https://doi.org/10.1016/j.ijpe.2013.03.001>

Liu, S., Tan, W., Xu, Z., & Liu, X. (2014). A tree search algorithm for the container loading problem. *Computers & Industrial Engineering*, *75*, 20–30. <https://doi.org/10.1016/j.cie.2014.05.024>

Pisinger, D. (2002). Heuristics for the container loading problem. *European Journal of Operational Research*, *141*(2), 382–392. [https://doi.org/10.1016/S0377-2217\(02\)00132-7](https://doi.org/10.1016/S0377-2217(02)00132-7)

Rimienė, K., & Grundey, D. (2007). Logistics Centre Concept through Evolution and Definition. *ENGINEERING ECONOMICS*, *54*(4), 87–95. Recuperado a partir de <https://inzeko.ktu.lt/index.php/EE/article/view/12286>

Sangaiah, A. K., Hosseinabadi, A. A. R., Shareh, M. B., Bozorgi Rad, S. Y., Zolfagharian, A., & Chilamkurti, N. (2020). IoT Resource Allocation and Optimization Based on Heuristic Algorithm. *Sensors*, *20*(2), 539. <https://doi.org/10.3390/s20020539>

Silva Semaan, G., de Moura Brito, J. A., Machado Coelho, I., Franco Silva, E., Cesar Fadel, A., Satoru Ochi, L., & Maculan, N. (2020). A Brief History of Heuristics: from Bounded Rationality to Intractability. *IEEE Latin America Transactions*, *18*(11), 1975–1986. <https://doi.org/10.1109/TLA.2020.9398639>

- Sommerville, I. (2011). *Ingeniería de Software* (9^a ed.). México: Pearson Educación.
- Szymonik, A. (2012). Information Technologies in Logistics. *Technical University*. Recuperado a partir de <https://www.researchgate.net/publication/296705505>
- Varajão, J., Trigo, A., Pereira, J. L., & Moura, I. (2022). Information systems project management success. *International Journal of Information Systems and Project Management*, 9(4), 62–74. <https://doi.org/10.12821/ijispm090404>
- Zhu, W., Oon, W.-C., Lim, A., & Weng, Y. (2012). The six elements to block-building approaches for the single container loading problem. *Applied Intelligence*, 37(3), 431–445. <https://doi.org/10.1007/s10489-012-0337-0>

ÍNDICE

1.	DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN	3
1.1.	DESCRIPCIÓN DE LA EMPRESA	3
1.2.	DESCRIPCIÓN DEL ÁREA DE ESTUDIO	3
1.3.	DESCRIPCIÓN DE LA PROBLEMÁTICA	3
2.	DEFINICIÓN PROYECTO	4
2.1.	OBJETIVOS DEL PROYECTO.....	4
3.	ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE.....	4
3.1.	ETAPA: PROYECTO BLASTOFF	4
3.2.	ETAPA: CAPTURA DE REQUISITOS.....	4
3.2.1.	<i>Alcances del Producto</i>	4
3.3.	ETAPA: ESCRITURA Y PROTOTIPOS DE REQUISITOS.....	5
3.3.1.	<i>Requerimientos funcionales</i>	5
3.3.2.	<i>Casos de Uso del Sistema</i>	5
3.3.3.	<i>Maquetas del Sistema (Mockups)</i>	6
4.	DISEÑO FÍSICO.....	7
4.1.	RECURSOS.....	7
4.1.1.	HARDWARE	7
4.1.2.	SOFTWARE	7



HOJA DE CONTROL

Organización	Alimentos del Pedregal		
Proyecto	Diseño y desarrollo de un algoritmo heurístico para la optimización de cajas sobre pallets para Alimentos del Pedregal		
Entregable	Especificación de Requerimientos		
Contacto	Jorge Gallego		
e-mail			
Cargo	Jefe de logística		
Versión/Edición	0002	Fecha Versión	27/06/2022
Aprobado por		Fecha Aprobación	DD/MM/AAAA

2

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
0001	Versión inicial	Gonzalo Basualto	21/06/2022
0002	Correcciones profesor guía	Anaís Gaete	27/06/2022

INTEGRANTES DEL GRUPO DE PROYECTO

Apellidos Nombres	Correo electrónico
Basualto Ulloa Gonzalo	gbasualto18@alumnos.utralca.cl
Gaete Veloso Anaís	agaete18@alumnos.utralca.cl

1. DEFINICIÓN DE LA EMPRESA O INSTITUCIÓN

1.1. Descripción de la empresa

Los antecedentes generales de la empresa son:

Nombre: Alimentos del Pedregal.

Rubro: Distribución y consumo masivo de productos alimenticios.

1.2. Descripción del área de estudio

El proyecto se desarrolla en el área de estudio relacionado a la logística y cadena de distribución de la empresa. Alimentos del Pedregal, al estar enfocados a la distribución masiva de productos alimenticios a diversas salas de ventas, tiene múltiples desafíos en las operaciones logísticas, una de ellas es la carga de cajas de productos en los pallets que se transportan en sus camiones.

1.3. Descripción de la problemática

Actualmente, Alimentos del Pedregal cuenta con una herramienta Excel, que no considera sus especificaciones, y con la que realizan la distribución de cajas sobre pallets, esto implica aumento de costos debido al uso ineficiente de recursos, como tiempo, espacio y personal humano, entre otros.

Para una empresa que cuenta con más de 40 salas de ventas distribuidas entre Copiapó y Puerto Montt, es necesaria la implementación de un sistema que le permita utilizar de manera eficiente los recursos antes mencionados.



2. DEFINICIÓN PROYECTO

2.1. Objetivos del proyecto

Desarrollar un algoritmo heurístico para optimizar el tiempo de respuesta y la organización de las cajas sobre pallets en tiempo real, según variables como cantidad de cajas, categorías, peso, dimensiones de las cajas y dimensiones del camión.



3. ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

3.1. Etapa: Proyecto Blastoff

La etapa de especificación de requerimientos comenzó con una reunión en dependencias de uno de los centros de distribución de Alimentos del Pedregal. En esta reunión, el Sr. Jorge Gallego, encargado del área de logística de los productos congelados, indicó la necesidad que tienen y qué quieren abordar con el desarrollo del proyecto. Estas son: mejorar los tiempos en la toma de decisiones para el armado de pallets, tarea que hoy se realiza manualmente y donde se apela principalmente a la experiencia de los operarios. Los principales riesgos del proyecto se pueden asociar a las dificultades de adaptación y adopción del sistema.

Posterior a esta reunión, el equipo de proyecto se dispone a realizar una propuesta y evaluarla en conjunto con la contraparte, para validar si efectivamente se abordan todas las necesidades del problema.

3.2. Etapa: Captura de requisitos.

3.2.1. Alcances del Producto

Diseño y desarrollo de un algoritmo heurístico para la optimización de las cajas sobre pallet.

Este algoritmo heurístico ordenará de manera óptima y rápida las cajas sobre los pallets, de manera que las personas a cargo no deban tomar decisiones sobre la posición de estas.

El algoritmo dará apoyo al área de logística de los productos congelados, ya que hará el proceso más expedito, permitirá que la persona a cargo del proceso de carga de pallet no deba evaluar la posición de cada caja, si no que venga dada, la heurística solo deberá indicarles a las personas que carguen las cajas cuáles y dónde ubicarlas.

3.3. Etapa: Escritura y prototipos de requisitos

3.3.1. Requerimientos funcionales

El sistema a desarrollar debe cumplir con los siguientes servicios:

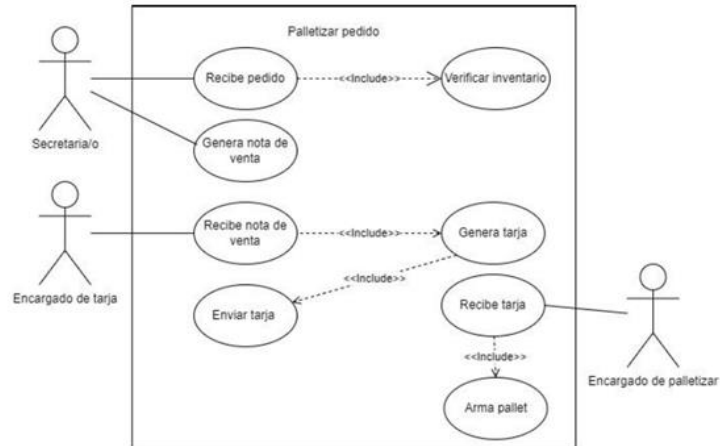
- Recibir como input (entrada) con el listado de productos a ordenar sobre el pallet.
- Generar una tarja, con el orden y ubicación de cada producto sobre el pallet.
- Tener las dimensiones de las cajas de cada producto previamente cargados para realizar la asignación.

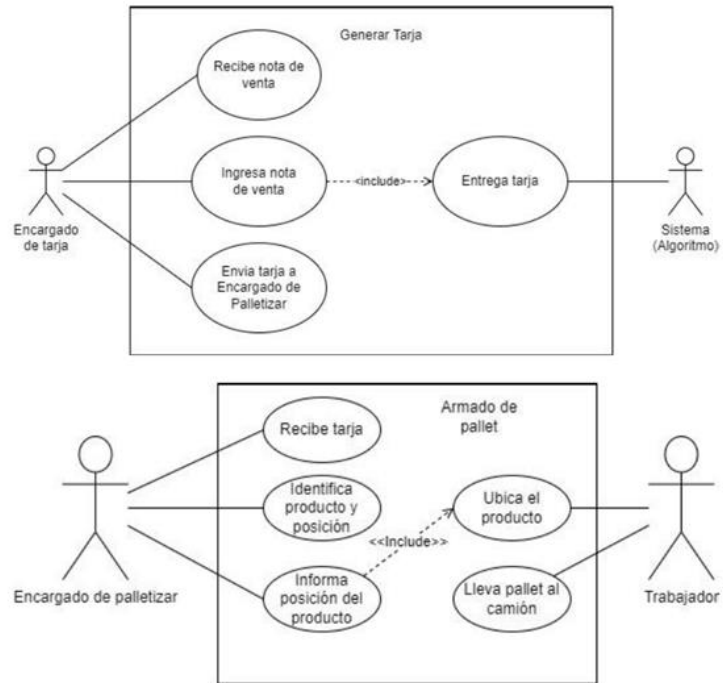


Todo lo anterior, debe considerar las siguientes restricciones:

- Peso de los productos (pesados más abajo, livianos más arriba).
- Dimensiones de pallet y altura máxima del camión.
- Orientación de las cajas (algunas cajas se podrían ubicar en posición horizontal, más no vertical).
- Peso máximo del pallet.

3.3.2. Casos de Uso del Sistema





3.3.3. Maquetas del Sistema (Mockups)

Cabe destacar que la complejidad de la interfaz gráfica es bastante acotada, puesto que el modelo matemático y la programación detrás de este es lo realmente importante.

Menú Principal



Menú Palletizar



7

4. DISEÑO FÍSICO

4.1. Recursos

4.1.1. Hardware

Los requisitos mínimos con los que debe contar el hardware a utilizar para llevar a cabo el proyecto es el siguiente:

Computador con un procesador Intel Core i3, de 1,5 Ghz en adelante, 4GB de memoria RAM, espacio en Disco Duro o Unidad de estado sólida (ideal) 1 GB.

4.1.2. Software

Sistema Operativo Windows 10

Microsoft Excel 365

Python 3.10

Anaconda Navigator 2.2.0