

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA ELECTRÓNICA Y  
AUTOMÁTICA INDUSTRIAL



**UNIVERSITAS**  
*Miguel Hernández*

"ESTUDIO DE OPTIMIZACIÓN DE LA  
PLANIFICACIÓN DE PROCESOS EN UNA  
PLANTA DE PRODUCCIÓN DE  
CALZADO"

TRABAJO FIN DE GRADO

Noviembre -2025

AUTOR: Almudena López Palomar

DIRECTOR/ES: Joaquín Sánchez Soriano

Juan Carlos Gonçalves Dosantos



## RESUMEN

Este trabajo aborda la optimización de la planificación y secuenciación en una planta de producción de calzado, teniendo en cuenta las rutas por tipo de producto, los tiempos de proceso y el número de máquinas por etapa. Se han desarrollado y comparado dos modelos matemáticos, uno con enfoques par a par y otro por bloques de producto, junto con una versión simplificada de este último. Además, se ha implementado una heurística basada en órdenes de producción y permutaciones que permite obtener soluciones en tiempos razonables. Los resultados muestran que, aunque los modelos ofrecen soluciones óptimas para demandas pequeñas, la heurística mejora significativamente la eficiencia en situaciones con alta demanda o numerosos productos, logrando una planificación realista del proceso productivo.



## **ABSTRACT**

This work addresses the optimization of planning and scheduling in a footwear production plant, taking into account product-specific routes, processing times, and the number of machines per stage. Two mathematical models have been developed and compared, one with a pair-by-pair approach and another based on product blocks, along with a simplified version of the latter. In addition, a heuristic based on production orders and permutations has been implemented, allowing solutions to be obtained within reasonable times. The results show that, while the models yield optimal solutions for small demands, the heuristic significantly improves efficiency in cases with high demand or many products, achieving a realistic planning of the production process.



## RESUM

Aquest treball aborda l'optimització de la planificació i seqüenciació en una planta de producció de calçat, tenint en compte les rutes per tipus de producte, els temps de procés i el nombre de màquines per etapa. S'han desenvolupat i comparat dos models matemàtics, un amb enfocaments parell a parell i un altre per blocs de producte, juntament amb una versió simplificada d'aquest últim. A més, s'ha implementat una heurística basada en ordres de producció i permutacions que permet obtenir solucions en temps raonables. Els resultats mostren que, encara que els models ofereixen solucions òptimes per a demandes xicotetes, l'heurística millora significativament l'eficiència en situacions amb alta demanda o nombrosos productes, aconseguint una planificació realista del procés productiu.



## ÍNDICE

Índice .....	6
Índice de tablas .....	8
Índice de ilustraciones .....	9
1. Introducción.....	10
1.1 La organización industrial y la planificación .....	10
1.2 Niveles de planificación .....	11
1.3 Problemas típicos de la planificación .....	12
1.4 Optimización, investigación operativa y modelos matemáticos .....	13
2. Estado del arte .....	16
2.1 Introducción.....	16
2.2 Revisión de la literatura en planificación y secuenciación de la producción: modelos y heurísticas .....	16
2.3 Limitaciones de la literatura y hueco de investigación.....	18
3. Descripción de la empresa y sistema productivo .....	20
3.1 Descripción de la planta de producción.....	20
3.2 Materias primas y flujo de entrada .....	20
3.3 Proceso productivo por etapas.....	21
3.4 Rutas de producción por tipo de zapato.....	27
3.5 Organización y recursos humanos .....	38
4. Modelos matemáticos y desarrollo del proyecto .....	42
4.1 Introducción al modelado .....	42
4.2 Primer modelo matemático: enfoque par a par.....	42
4.3 Segundo modelo matemático: por bloques de productos .....	49
4.4 Simplificación del segundo modelo matemático: por bloques con una máquina por etapa .....	58
4.5 Simulación heurística basada en órdenes y permutaciones .....	61

5. Conclusiones.....	70
Bibliografía.....	72
Apéndice. Código Python de la heurística.....	74



## ÍNDICE DE TABLAS

Tabla 1 .....	48
Tabla 2 .....	57
Tabla 3 .....	58
Tabla 4 .....	59
Tabla 5 .....	60
Tabla 6 .....	61
Tabla 7 .....	67
Tabla 8 .....	68
Tabla 9 .....	69



## ÍNDICE DE ILUSTRACIONES

Ilustración 1 .....	22
Ilustración 2 .....	23



## 1. INTRODUCCIÓN

El sector del calzado en España siempre ha tenido importancia a nivel nacional e internacional, en especial en la Comunidad Valenciana, que representa dos terceras partes de la producción española. En concreto, en la provincia de Alicante se concentra el 64% de la producción total del país, y, por ende, aquí se encuentran muchas de las empresas más representativas del sector (Mora y de Lucas, 2017).

Teniendo una relevancia tan clara, cabría esperar que estas empresas estuvieran dotadas de una buena organización industrial, pero la realidad es que no siempre es así, y muchas veces la coordinación de procesos y recursos todavía deja margen de mejora. Según Cabral (2022), “la organización industrial es el estudio del funcionamiento de los mercados e industrias, y en particular, de la manera como las empresas compiten entre ellas” (p. 19).

En este trabajo nos centraremos en el funcionamiento industrial, prestando especial atención a las distintas etapas del proceso productivo. La planificación es una parte muy importante, ya que afecta directamente a los costes, al tiempo de producción y al uso de los recursos. Hoy en día, con tanta competencia a nivel global, las empresas necesitan organizarse de la mejor manera posible si quieren seguir siendo eficientes, ofrecer calidad y adaptarse al mercado.

### 1.1 LA ORGANIZACIÓN INDUSTRIAL Y LA PLANIFICACIÓN

La organización industrial tiene un papel clave dentro de cualquier empresa productiva, ya que se encarga de coordinar todas las actividades necesarias para transformar materias primas en productos terminados de forma eficiente. No se trata solo de fabricar, sino de hacerlo con una buena gestión del tiempo, los recursos y las personas.

En las plantas de producción, el objetivo principal es lograr un equilibrio entre la calidad, el coste y el plazo de entrega. Para conseguirlo, la planificación se convierte en una herramienta esencial.

La planificación permite organizar la producción de forma que se decida con antelación qué se va a fabricar, en qué momento, con qué recursos y en qué orden. Según Heizer, Render y Parra (2007), este proceso ayuda a coordinar las distintas áreas de la empresa (como producción, compras o logística), aprovechando mejor los recursos disponibles y evitando retrasos o tiempos muertos. De esta manera, se consigue un funcionamiento más eficiente y ordenado de toda la planta.

En la práctica, planificar implica enfrentarse a muchas variables: la demanda de los clientes, la disponibilidad de materiales, la capacidad de las máquinas, los turnos del personal o incluso los imprevistos que surgen durante la producción. Cuando estas variables no se controlan bien, aparecen los problemas típicos de las fábricas: demoras en la entrega, acumulación de inventario, sobrecostes, fallos de coordinación o periodos en los que unas máquinas están saturadas mientras otras están paradas.

Por ello, muchas empresas, con el objetivo de mejorar la planificación, recurren a métodos de optimización. Estas herramientas, basadas en modelos matemáticos, simulaciones o algoritmos, permiten analizar distintas alternativas y encontrar la forma más eficiente de organizar el trabajo. En lugar de depender únicamente de la experiencia o la intuición, la optimización ofrece una base más objetiva para tomar decisiones, reduciendo costes y mejorando la productividad.

En este punto, resulta interesante profundizar en cómo se estructura la planificación dentro de la empresa, ya que no todas las decisiones se toman con el mismo alcance ni en el mismo plazo temporal. Esto nos lleva a hablar de los distintos niveles de planificación.

## 1.2 NIVELES DE PLANIFICACIÓN

En una empresa industrial, la planificación se organiza en diferentes niveles según el tipo de decisiones que se tomen y el horizonte temporal considerado. De forma general, se distinguen tres niveles: estratégico, táctico y operativo (Los Santos, 2010). Cada uno de ellos cumple un papel clave para que la producción funcione de manera coordinada y eficiente.

El nivel superior es el nivel estratégico, y su horizonte temporal suele abarcar varios años. En él se toman decisiones que determinarán la dirección de la empresa a largo plazo. Estas decisiones están relacionadas con la gestión global del negocio, como la ubicación de nuevas fábricas, la inversión en maquinaria, el lanzamiento de nuevos productos o las políticas de sostenibilidad. El objetivo principal es definir la visión y el recorrido futuro de la organización.

Por otro lado, el nivel táctico tiene un horizonte temporal más corto, normalmente de meses o trimestres. Aquí se planifican los recursos, la capacidad de producción, el personal o los materiales necesarios para cumplir con los objetivos establecidos en el nivel estratégico. En este sentido, el nivel táctico actúa como un puente entre la estrategia general de la empresa y las acciones concretas del día a día.

Por último, en el nivel operativo se definen las actividades diarias que permiten llevar a cabo la producción de forma ordenada. Este nivel se centra en la ejecución concreta en el taller o la planta, ocupándose de aspectos como la programación diaria, la secuenciación de tareas, la asignación de máquinas o el control del inventario.

En resumen, los tres niveles de planificación están estrechamente relacionados y resultan imprescindibles para el correcto funcionamiento de la empresa. En este trabajo, el foco se sitúa principalmente en los niveles táctico y, sobre todo, operativo, ya que es en ellos donde se lleva a cabo la planificación y secuenciación de la producción mediante modelos matemáticos, simulación y algoritmos.

### 1.3 PROBLEMAS TÍPICOS DE LA PLANIFICACIÓN

Aunque una empresa pueda tener una buena organización y planificación, en la práctica suelen aparecer algunos problemas que afectan al día a día de la producción (MacCarthy, 2006). Los más comunes son los retrasos, los sobrecostos, la descoordinación entre departamentos y el desperdicio de recursos, como tiempo de máquina o inventario acumulado.

Estas dificultades suelen aparecer por varias causas: falta de información o datos incompletos, mala secuenciación de tareas, falta de sincronización entre las diferentes etapas de producción o incluso imprevistos como averías, retrasos de proveedores (por ejemplo, en la entrega de materias primas) o cambios inesperados en la demanda (Chankov, Hütt, & Bendul, 2016).

Todos estos problemas acaban afectando al funcionamiento general de la fábrica: se pierde productividad, suben los costes y pueden verse afectados la calidad o los plazos de entrega al cliente. Por eso, muchas empresas empiezan a usar herramientas de optimización, simulación y algoritmos, que les ayudan a planificar mejor y reducir estos problemas, asegurándose de que la producción fluya de forma continua y que los recursos se usen de forma más eficiente.

#### 1.4 OPTIMIZACIÓN, INVESTIGACIÓN OPERATIVA Y MODELOS MATEMÁTICOS

La palabra optimización, dentro de un entorno industrial, significa conseguir los mejores resultados de la manera más eficiente posible. Es decir, aprovechar al máximo los recursos disponibles, como el tiempo, la maquinaria, el personal o las materias primas, para alcanzar los objetivos de producción. Básicamente, se trata de encontrar un buen equilibrio entre coste, calidad y tiempo, para que la empresa sea más competitiva y sostenible a largo plazo.

Para conseguirlo, muchas empresas utilizan modelos de optimización y herramientas digitales que les ayudan a tomar decisiones basadas en datos. Según Casiello (2021), estas herramientas permiten probar distintas maneras de organizar la producción y ver cuál funciona mejor, por ejemplo, reduciendo tiempos de espera o aprovechando más las máquinas. También se pueden usar métodos heurísticos o de simulación para estudiar el comportamiento del sistema ante distintos escenarios, como cambios en la demanda.

Este tipo de análisis se enmarca en la investigación operativa, la disciplina que ayuda a tomar decisiones más inteligentes en entornos complejos. Según Hillier y Lieberman

(2010), se centra en usar métodos analíticos para encontrar la mejor manera de hacer las cosas, sobre todo cuando hay recursos limitados y muchas variables que deben ser tenidas en cuenta. En otras palabras, sirve para buscar soluciones eficientes a problemas complejos, y es justo lo que hace falta al organizar la producción de una fábrica.

Dentro de la investigación operativa, los modelos matemáticos son la herramienta principal. Nos ayudan a ver de forma clara cómo funciona la producción: qué hace cada máquina, cómo se reparte el trabajo entre el personal, cómo llegan los materiales y qué límites tenemos que respetar. Con estos modelos, es posible efectuar pruebas de las distintas formas de organizar la producción y ver cuál es la más eficiente antes de fabricar un solo zapato en la planta.

Una parte importante que cabe mencionar de los modelos matemáticos es la programación matemática, que nos permite encontrar soluciones óptimas respetando las restricciones. Winston (2004) explica dos enfoques principales: los métodos exactos, que dan la solución óptima pero a veces tardan mucho en problemas grandes, y los métodos heurísticos, que no aseguran la solución óptima pero permiten obtener resultados bastante buenos mucho más rápido.

En el presente Trabajo de Fin de Grado se aplica la optimización a una hipotética planta de producción de calzado que se ha diseñado de la forma más realista posible utilizando conversaciones con profesionales del calzado. El objetivo es desarrollar un modelo que permita planificar y secuenciar la producción teniendo en cuenta las diferentes etapas del proceso, las distintas rutas de los zapatos y los tiempos de cada tipo de producto.

Para ello, se emplea el lenguaje de programación Python para el desarrollo completo del proyecto, tanto en la formulación y resolución del modelo de optimización mediante el software Gurobi, como en la implementación del programa heurístico correspondiente. Los programas desarrollados consideran diversos factores relevantes del proceso productivo, entre ellos: la demanda de los productos, las etapas de producción, la

disponibilidad de máquinas, la ruta específica asociada a cada tipo de calzado y los tiempos de procesamiento de cada operación.

Esta configuración permite planificar y secuenciar la producción de manera eficiente, así como analizar el comportamiento del sistema bajo diferentes combinaciones de los factores mencionados (Gurobi Optimization, LLC, 2024; Python Software Foundation, 2024).

Para la ejecución de las simulaciones y experimentos computacionales, se ha utilizado un ordenador portátil personal con características técnicas suficientes para el desarrollo de este tipo de tareas. El equipo dispone de un procesador Intel Core i5-7200U con una frecuencia de 2.50 GHz, 8 GB de memoria RAM, un disco SSD de 466 GB y un disco duro (HDD) de 932 GB. Asimismo, cuenta con dos tarjetas gráficas: una NVIDIA GeForce GTX 950M con 2 GB de memoria dedicada y una Intel HD Graphics 620 integrada.

El sistema operativo empleado es de 64 bits, lo que garantiza la compatibilidad y el rendimiento adecuados para la ejecución de los programas desarrollados en Python y el entorno de optimización Gurobi.

## **2. ESTADO DEL ARTE**

### **2.1 INTRODUCCIÓN**

El estado del arte permite situar el Trabajo de Fin de Grado dentro del contexto general de los estudios sobre la planificación de la producción en entornos industriales. Dado que en este proyecto se desarrollan varios modelos matemáticos y, posteriormente, un método heurístico, es necesario revisar la literatura técnica para identificar qué enfoques existen y qué herramientas se utilizan.

El objetivo es presentar una visión general del marco en el que se encuadra el problema: la planificación en una planta con varias etapas, la existencia de rutas distintas según el producto, la asignación de máquinas y los tiempos de proceso. Estos elementos que son comunes en muchas fábricas sirven para contextualizar qué herramientas de optimización resultan útiles.

A partir de esta idea, el estado del arte se centra en explicar varios artículos que guardan relación con este trabajo, tanto de producción en general como de estudios más cercanos al sector del calzado.

### **2.2 REVISIÓN DE LA LITERATURA EN PLANIFICACIÓN Y SECUENCIACIÓN DE LA PRODUCCIÓN: MODELOS Y HEURÍSTICAS**

La literatura sobre planificación y secuenciación integra varias perspectivas para organizar la producción de manera realista. Los modelos matemáticos se utilizan para representar el sistema junto con sus restricciones, y por otro lado, las heurísticas sirven para obtener soluciones más rápidas cuando el problema crece y el modelo exacto no es computacionalmente factible. Este planteamiento combinado aparece tanto en estudios generales de producción como en investigaciones centradas en la industria del calzado. A continuación se describen tres artículos relacionados con este trabajo.

El primer artículo es el de Süer, Subramanian y Huang (2009), que trabaja con una empresa del ámbito del calzado. Los autores definen un modelo matemático que asigna

productos a diferentes celdas de trabajo y programa su secuencia, teniendo en cuenta la capacidad disponible y el orden de las operaciones. Su objetivo principal es equilibrar la carga de las celdas para evitar cuellos de botella. Además del modelo, el estudio incluye varios procedimientos heurísticos para generar soluciones cuando la formulación se vuelve demasiado grande o compleja. Con esta combinación de herramientas se pueden analizar los cambios en los resultados según la carga, el número de familias de productos o la secuencia final.

El segundo trabajo, de Saraçoğlu, Süer y Gannon (2021), se centra en una fábrica de calzado con varias etapas en paralelo. Aquí los autores desarrollan un modelo matemático cuyo objetivo es reducir el makespan<sup>1</sup> y el tiempo total de flujo o flowtime. Consideran una metodología dividida en tres fases: primero asignan la mano de obra disponible a las celdas, después realizan la carga de productos en esas celdas combinando el modelo con un algoritmo genético y, para finalizar, programan la secuencia con un esquema de flowshop. Es decir, en un flowshop todos los productos hacen las mismas etapas en el mismo orden. Así pueden analizar el rendimiento según la distribución del trabajo y el equilibrio entre etapas.

Por último, el estudio de Oksuz, Buyukozkan y Satoglu (2017) aborda un problema de asignación de operarios y balanceo de una línea en U. Aunque no pertenece al sector del calzado, el planteamiento se puede aplicar a problemas de planificación y asignación de recursos. Los autores desarrollan, además del modelo, dos metaheurísticas para mejorar la eficiencia de la línea y reducir tiempos improductivos, lo que demuestra que ambos métodos pueden trabajar juntos para gestionar sistemas productivos complejos.

Además de estos artículos, conviene mencionar de manera resumida otros trabajos de interés.

---

<sup>1</sup> Tiempo total requerido para completar la totalidad de los trabajos o tareas programadas.

Meilanitasari y Shin (2021) realizan una revisión amplia sobre la combinación de métodos de predicción y técnicas de optimización para planificar la secuencia de producción en sistemas flexibles tipo job shop.

Por otro lado, en el estudio de Muliono, Silviana y Novita (2024) se aplican algoritmos genéticos a una fábrica de calzado y se comparan varios métodos de selección para ver qué programación es más eficiente y reduce mejor los tiempos de producción.

Para concluir, Rezig, Ezzeddine, Turki y Rezg (2020) proponen un modelo matemático que optimiza el plan de producción en sistemas de eventos discretos, integrando restricciones operativas y mantenimiento preventivo dentro de la misma formulación.

En resumen, el uso de modelos matemáticos y heurísticas permite planificar y secuenciar la producción de manera eficiente, sirviendo de base para este Trabajo de Fin de Grado.

### 2.3 LIMITACIONES DE LA LITERATURA Y HUECO DE INVESTIGACIÓN

Aunque los estudios aportan ideas valiosas sobre planificación y secuenciación, presentan limitaciones si se comparan con la complejidad de una planta real de calzado. En primer lugar, suelen trabajar con plantas simplificadas, con pocas celdas o etapas consecutivas, muy alejadas de los 20–30 procesos que pueden tener las fábricas reales. Esto hace que sus rutas de productos sean más simples y no contemplen saltos de etapa ni combinaciones complejas de procesos.

Por otro lado, la integración entre planificación, secuenciación y disponibilidad de máquinas no es completa. Por ejemplo, Rezig et al. (2020) incluyen la disponibilidad de máquinas y el mantenimiento en un modelo matemático, pero no combinan heurística ni simulación para rutas complejas. De manera similar, Muliono et al. (2024) aplican algoritmos genéticos para optimizar la secuencia de producción en calzado, pero no consideran la planificación de múltiples etapas ni rutas no consecutivas.

Las heurísticas que se usan en estos estudios también son limitadas. Se aplican algoritmos tradicionales o genéticos, pero no hay aproximaciones basadas en órdenes de producción y permutaciones evaluadas para manejar rutas reales complejas. Además, no se analiza la escalabilidad: los trabajos existentes no prueban casos con muchos productos y etapas, y sus soluciones no se han validado en escenarios grandes.

Finalmente, el tipo de planificación que presentan no refleja fielmente el flow-shop flexible característico de la industria del calzado. Si bien Meilanitasari y Shin (2021) destacan la importancia de sistemas tipo job-shop flexible y la secuencia de productos, no muestran un caso industrial aplicado que integre todos los elementos que un sistema real exige.

A partir de las limitaciones identificadas, se observa un hueco en la literatura que este TFG busca cubrir. Algunos estudios incluyen modelos matemáticos y heurísticas, pero ninguno implementa de manera práctica en un entorno de programación una planta de calzado con rutas distintas y múltiples etapas.

Tampoco hay heurísticas diseñadas para evaluar rutas complejas. Por eso, este trabajo propone una heurística basada en órdenes de producción y permutaciones que, además de generar y evaluar diferentes órdenes de producción, incorpora una búsqueda tabú para evitar repetir movimientos recientes y un criterio de parada por estancamiento, lo que permite explorar soluciones de manera más ordenada y eficiente.

Además, los artículos revisados simplifican mucho el proceso productivo porque usan pocas etapas, pocas celdas y estructuras muy ordenadas. Este TFG quiere reflejar de manera realista la planta de producción, y eso significa, considerar muchas más etapas y rutas no consecutivas, para acercarnos a la forma de producción habitual en la industria del calzado.

Teniendo en cuenta estas carencias, el siguiente paso es presentar la planta de producción ficticia utilizada para el estudio.

### **3. DESCRIPCIÓN DE LA EMPRESA Y SISTEMA PRODUCTIVO**

#### **3.1 DESCRIPCIÓN DE LA PLANTA DE PRODUCCIÓN**

El tipo de empresa teórica que se presenta en este TFG es una planta de producción dedicada al calzado de mujer. Está diseñada para parecer lo más realista posible, basándome en conversaciones con profesionales del sector y en el funcionamiento de este tipo de fábricas.

En cuanto a su capacidad productiva, la empresa no corresponde a una gran corporación, sino a una planta de tamaño medio orientada al cuidado del producto y a la optimización de los procesos de fabricación. El objetivo principal no es la producción en masa, sino la elaboración de calzado de alta calidad, priorizando la atención al detalle y el control en cada etapa del proceso productivo.

El objetivo de la planta es fabricar distintos modelos de zapatos de mujer, la mayoría elaborados con piel de vacuno por su resistencia y acabado. Cada modelo sigue su propia ruta dentro del proceso productivo, compartiendo algunas etapas con otros tipos de calzado, aunque no todas sean las mismas.

La organización de la planta se ha planteado de forma lógica y sencilla, de manera que refleje la estructura de los procesos en una fábrica real. A partir de aquí, se describe con más detalle la estructura de la producción, las etapas y las máquinas que intervienen en cada una.

#### **3.2 MATERIAS PRIMAS Y FLUJO DE ENTRADA**

Las materias primas que se usan en la planta vienen de diferentes proveedores dentro del sector del calzado. No es relevante entrar en detalle sobre los tiempos de entrega porque en este trabajo se da por hecho que todos los materiales están disponibles cuando empieza la producción. El objetivo es centrarse en cómo se organiza la fabricación dentro de la planta, no en la parte de transporte o gestión de stock. Por eso, se asume que siempre hay materiales suficientes para trabajar sin interrupciones.

En cuanto a los tipos de materiales utilizados, la principal materia prima es la piel, necesaria para el corte y forrado, así como otros elementos como la espuma de látex, el tope, el contrafuerte y la planta de montado. También se utilizan suelas, plantillas, hilos, cordones y algunos componentes metálicos, además del material de embalaje (cajas, papel, pernitos o separadores). Todos estos materiales llegan ya preparados desde las empresas proveedoras. Un mismo proveedor puede suministrar varios de ellos, como suelas y plantillas.

Con los materiales definidos y disponibles, el siguiente paso es describir cómo se transforman dentro de la planta a través de las distintas etapas del proceso productivo.

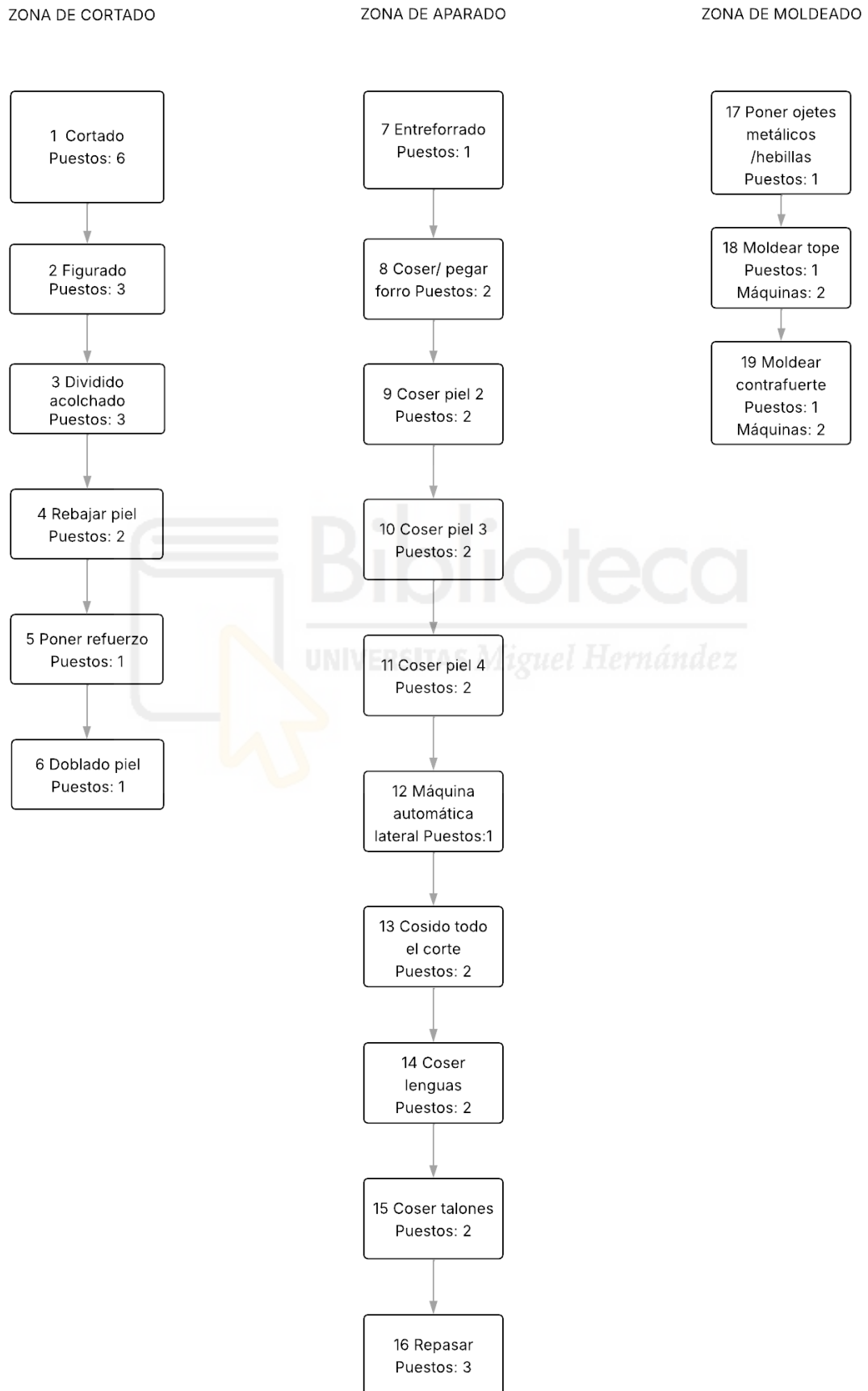
### 3.3 PROCESO PRODUCTIVO POR ETAPAS

Antes de pasar a la descripción de todas las etapas que componen la producción de los distintos tipos de zapatos, cabe destacar que existe un paso previo de diseño. En esta fase, los diseñadores de calzado elaboran la ficha técnica del modelo, donde se detallan todos los aspectos necesarios para su posterior fabricación: materiales, tipo de suela, componentes metálicos, cordones y demás características que darán forma al zapato final.

Dentro de la planta de producción nos encontramos con cinco zonas. Se pueden clasificar en cortado, aparado, moldeado, ensamblaje y acabado. Cada zona se encarga de unas etapas concretas, con unas máquinas específicas y operarios especializados.

### Ilustración 1

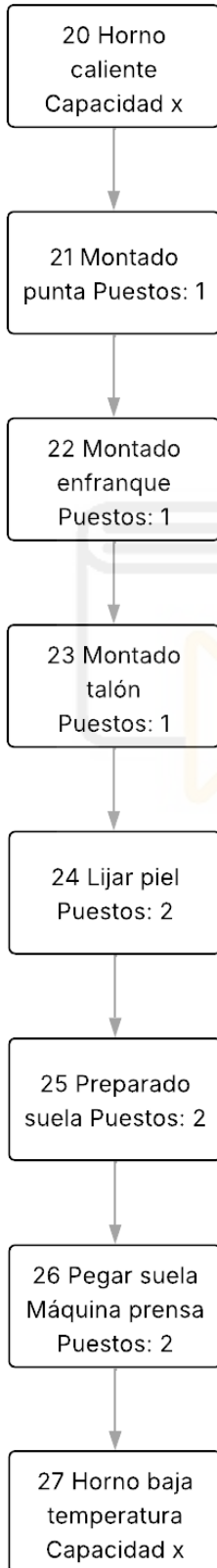
Etapas de las primeras tres zonas de la planta de producción



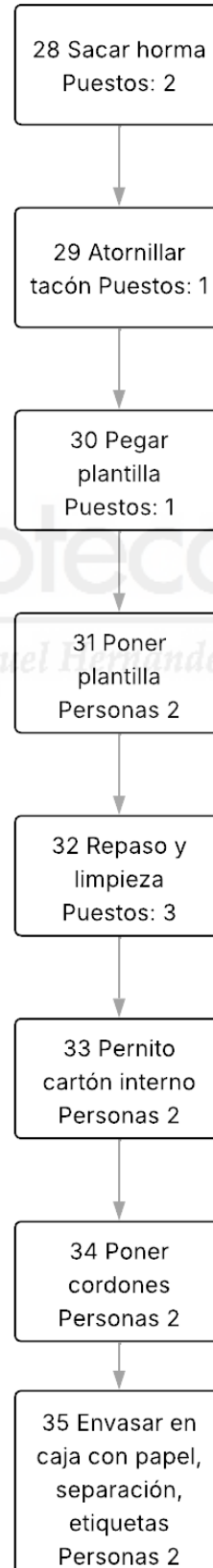
## Ilustración 2

Etapas de las últimas dos zonas de la planta de producción

### ZONA DE ENSAMBLAJE



### ZONA DE ACABADO



Empezando por la zona de cortado, aquí es donde se preparan todas las piezas que formarán el zapato. Podemos distinguir seis etapas:

-Cortado: en esta etapa se realiza el corte de las pieles y forros necesarios. Se utiliza una máquina de corte con unos troqueles de acero que tienen el patrón con la forma exacta.

-Figurado: una vez las piezas están cortadas, se marcan con un color que se vea en la piel para facilitar a la hora de coser.

-Dividido del acolchado: con una máquina de dividir se divide el acolchado que llevarán algunos zapatos, sobre todo deportivos y botas, para mejorar la comodidad.

-Rebajar piel: en este proceso se afinan los bordes de las piezas para que después sea más fácil doblarlas o coserlas con una máquina rebajadora. Consiguiendo así que las uniones queden más limpias y con un mejor acabado.

-Poner refuerzo: se aplica un refuerzo especial en algunas zonas del zapato, sobre todo en modelos más delicados o con tacón. Este refuerzo ayuda a mantener la forma y da más consistencia al conjunto.

-Doblado piel: finalmente, con una máquina dobladora se doblan los bordes rebajados de la piel hacia el interior para dejar un acabado más limpio y resistente antes de pasar al aparado.

Una vez se tienen las piezas cortadas, el siguiente paso es el aparado. En estas etapas se da forma a la parte superior del zapato, uniendo las diversas piezas de piel y forro que se prepararon en el cortado. Es una de las partes más delicadas del proceso, ya que combina trabajo manual con distintas máquinas de coser. Al mismo tiempo, según el tipo de calzado, algunas operaciones pueden variar, ya que no todos los modelos se aparán igual ni utilizan las mismas máquinas.

Las etapas que forman parte de esta zona son las siguientes:

-Entreforrado: se introducen los refuerzos interiores entre la piel y el forro. Esto ayuda a mantener mejor la forma del zapato, especialmente en el talón o empeine.

-Coser o pegar forro: dependiendo del tipo de zapato, el forro se puede coser o pegar a las piezas principales de piel. En los modelos de alta gama, suele pegarse pero en otros se cose para aportar resistencia.

-Coser piel: conforma las etapas 9, 10 y 11, en las que se cosen las piezas de piel que dan la forma general al zapato. Dependiendo del modelo, se necesitan máquinas distintas, como las de columna, planas o de brazo.

-Máquina automática lateral: esta máquina se utiliza para coser los laterales del zapato cuando este es un deportivo o tiene líneas rectas.

-Cosido de todo el corte: en esta etapa se ensamblan todas las piezas cosidas anteriormente, formando el esqueleto superior del calzado. Se suele usar una máquina de coser plana industrial.

-Coser lenguas: se cosen las lengüetas a la parte delantera del corte, vigilando que queden bien centradas y con el espacio justo para el montado.

-Coser talones: se cierra la parte trasera del zapato uniendo los extremos del corte. Es una costura importante porque define la forma final del talón y da firmeza a la pieza.

-Reparar: finalmente, a mano, con unas tijeras, las repasadoras revisan todas las costuras y eliminan restos de hilos o pequeñas imperfecciones. En algunos casos, se limpian los bordes o se planchan las piezas para dejarlo todo listo antes de pasar al moldeado.

Por la zona de moldeado no pasan todos los modelos, pero los que sí lo hacen es porque requieren de un refuerzo adicional o algún elemento decorativo. Hay tres etapas a destacar:

-Poner ojetes metálicos: en esta etapa se añaden los componentes metálicos al zapato, ya sean los ojetes por donde pasarán los cordones, o detalles de la marca que añaden valor al diseño. Se usa una máquina eléctrica de ojetes con pedal.

-Moldear tope: consiste en añadir una pequeña pieza que refuerza la parte delantera. La máquina ejerce calor y presión sobre la piel ya cosida.

-Moldear contrafuerte: al igual que con el tope, pero en la zona del talón, se coloca una pieza plástica que se adhiere gracias a la acción de la máquina de moldear contrafuerte.

En la zona de ensamblaje se incorporan todas las piezas que se han ido preparando en etapas pasadas sobre la horma para darle la forma en tres dimensiones al zapato. Se trabaja directamente sobre la horma, para ajustar la piel, pegar la suela... Son unas etapas relevantes, ya que la precisión al montarlo repercutirá en la calidad del acabado final.

-Horno caliente: antes de empezar a montar, se pasa el corte por un horno de calor. El objetivo es ablandar la piel para que sea más fácil adaptarla a la forma de la horma sin que se arrugue o se dañe.

-Montado de punta: con una máquina de montar la puntera, se encaja la parte delantera del zapato sobre la horma. Esta máquina estira y fija la piel para que quede tirante y con la forma correcta.

-Montado de enfranque: este proceso ajusta la parte media del zapato (entre la punta y el talón). Se usa una máquina de montar enfranques, similar a la etapa anterior, esta máquina acomoda la piel sobre la forma de manera que no queden pliegues.

-Montado de talón: aquí se coloca y tensa la parte trasera. Se emplea una máquina de montar talones, que estira la piel y asegura que quede alineada con el resto del zapato.

-Lijado de piel: ahora que tenemos la piel ajustada a la horma, se lija la base con una lijadora de rodillos. Esto sirve para igualar la superficie donde irá pegada la suela y eliminar restos de adhesivo o piel sobrante.

-Preparado de suela: por otro lado se va preparando la suela, se limpia y aplica cola para dejarla lista para pegarla. Este proceso se realiza con una máquina encoladora.

-Pegado de suela: con la máquina de pegar suelas se unen ambas partes. Se pone primero la suela y se ejerce presión contra la horma para que quede bien incorporada.

-Horno de baja temperatura: para terminar, se pasa por un horno suave que fija completamente el adhesivo y asienta las piezas. Esto da solidez al zapato para que quede firme y duradero.

Para finalizar, hay que hablar de la zona de acabado y envasado, donde se revisa, se limpia y se deja listo para enviar al cliente. Es una zona en la que predomina el trabajo manual, ya que muchas tareas se hacen a mano, aunque las primeras etapas todavía requieren la ayuda de algunas máquinas. Cada modelo pasa solo por las etapas que le correspondan según su tipo: no todos llevan tacón, ni plantilla pegada o cordones.

-Sacar horma: cuando el zapato está totalmente montado y el pegamento bien seco, se saca la horma con una máquina que ejerce presión desde la punta. Este paso requiere cuidado para no deformar el producto ni dañar la estructura conseguida durante el ensamblaje.

-Atornillar tacón: en los modelos que lo llevan, se coloca el tacón con una máquina atornilladora, que asegura la fijación. Después se revisa que quede alineado.

-Pegar plantilla: en los zapatos con tacón o plantilla a la vista, se aplica cola y se pega la plantilla interior, asegurando que quede bien centrada y sin arrugas.

-Poner plantilla: en los modelos más deportivos o cerrados, se introduce la plantilla directamente sin pegarla, para que pueda sacarse o sustituirse sin dificultad.

-Repaso y limpieza: aquí se revisa el zapato completo. Se eliminan restos de cola, se limpian todas las superficies y se retocan pequeños detalles para que quede impecable.

-Colocar pernito: se introduce un pernito normalmente de cartón que ayuda a mantener la forma del zapato. Se suele poner antes del envasado, sobre todo, si el calzado lleva cordones.

-Poner cordones: en los modelos que los incluyen, se colocan los cordones y se dejan ya pasados, de forma que el zapato quede listo para su presentación o para meterlo en la caja.

-Envasado: por último, cada par se coloca en su caja de la manera correspondiente, con el papel protector y separadores si necesita. Se añaden las etiquetas con la información del modelo y talla, y así, el producto queda completo para su distribución.

### 3.4 RUTAS DE PRODUCCIÓN POR TIPO DE ZAPATO

En nuestro proceso productivo se fabrican distintos modelos de calzado: desde botas y botines para la temporada otoño-invierno, hasta sandalias para la primavera-verano, pasando por deportivos. Cada uno de estos productos sigue una ruta diferente dentro de la planta, aunque, como es lógico, los modelos similares comparten parte del recorrido. La mayor distinción suele estar en los materiales empleados y en los acabados o colores.

En este apartado se describen las rutas completas de cada producto, contando de forma sencilla el recorrido que hace dentro de la planta y añadiendo al final la secuencia

numérica de las etapas. Así se entiende fácilmente cómo avanza cada modelo a lo largo del proceso, desde que se corta la piel hasta que sale terminado y envasado.

Para fabricar el producto 1, el deportivo de piel, primero se cortan y figuran las piezas de piel y forro. También se divide el acolchado que irá entre ambas capas y se rebajan los bordes de la piel para facilitar el cosido. En la zona de aparado se colocan los refuerzos entre piel y forro, se cosen las distintas piezas y se termina de cerrar el corte con la máquina automática lateral. Después se rematan detalles como lenguas y talones y se repasa el conjunto.

En la zona de moldeado se colocan los ojetes metálicos y se moldean el tope y el contrafuerte. Tras pasar por el horno caliente, se monta el zapato sobre la horma empezando por la punta y terminando por el talón. A continuación, en el ensamblaje, se liján las superficies, se prepara la suela y se pega con la prensa, pasando luego por el horno de baja temperatura para asegurar la unión.

Finalmente, se retira la horma, se coloca la plantilla, se repasa y limpia el zapato, se introducen los pernitos de cartón, se ponen los cordones y se envasa.

Ruta: 1 ⇒ 2 ⇒ 3 ⇒ 4 ⇒ 7 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 11 ⇒ 12 ⇒ 13 ⇒ 14 ⇒ 15 ⇒ 16 ⇒ 17 ⇒ 18 ⇒ 19 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 23 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 34 ⇒ 35

El producto 2 es un zapato de tacón fino y cerrado. Para fabricarlo, primero se cortan y figuran las piezas de piel, se colocan los refuerzos y se doblan los bordes para darles mejor acabado. Posteriormente, en el aparado, se unen las partes del talón y se introduce el forro con sus refuerzos interiores. Una vez cosidas y repasadas las piezas, se pasa a moldeado, donde se forman el tope y el contrafuerte, se calienta en el horno y se monta el zapato sobre la horma (punta, enfranque y talón).

En el ensamblaje, se lija la piel, se prepara y pega la suela con la prensa y se pasa por el horno de baja temperatura para asegurar la adhesión. Finalmente, se saca la horma y se atornilla el tacón. A continuación, se pega la plantilla y se limpia el zapato. Después, se coloca el pernito de cartón para mantener la forma y, por último, se envasa con su correspondiente caja y etiquetado.

Ruta: 1 ⇒ 2 ⇒ 4 ⇒ 5 ⇒ 6 ⇒ 15 ⇒ 7 ⇒ 8 ⇒ 16 ⇒ 18 ⇒ 19 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 23 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 29 ⇒ 30 ⇒ 32 ⇒ 33 ⇒ 35

El producto 3 es una sandalia con tacón. Es un modelo que requiere menos materia prima y se fabrica más rápido en comparación con los anteriores, algo habitual en el calzado de verano. Se empieza cortando y figurando las pieles. Luego se rebajan para que sean más fáciles de manipular, se colocan los refuerzos y se doblan las piezas para facilitar el armado. Después, se pegan los forros, se cosen las pieles y se pasa al repasado.

En la zona de ensamblaje, se monta el enfranque y se lija la piel. Por otro lado, se prepara la suela, que lleva el tacón incorporado, para pegarla con la máquina de prensa, y finalmente se lleva al horno de baja temperatura para darle firmeza.

Ya por último, en la zona de acabado, se saca de la horma y se pega la plantilla, que en este caso queda a la vista. Se repasa y limpia el zapato antes de colocar el pernito de cartón y envasarlo en su caja con papel y las correspondientes etiquetas.

Ruta: 1 ⇒ 2 ⇒ 4 ⇒ 5 ⇒ 6 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 16 ⇒ 22 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 30 ⇒ 32 ⇒ 33 ⇒ 35

El producto 4 es una sandalia de piel con una suela con plataforma. Es un modelo sencillo pero con un diseño algo más elaborado en la parte superior, ya que en el cortado se hacen unos pequeños cortes en las tiras principales para que puedan pasar otras tiras por dentro

sin necesidad de coserlas. Después de eso, se figuran las pieles, se rebaja la piel para facilitar su manipulación y se colocan los refuerzos necesarios.

En el aparado, se pegan los forros y se cosen algunas de las tiras, pasando después al repasado para comprobar que las uniones quedan limpias y bien acabadas.

Más adelante, en el ensamblaje, se monta el enfranque, se lija la piel y se prepara la suela antes de pegarla con la máquina de prensa. Luego se pasa por el horno de baja temperatura para asegurar que todas las piezas queden bien selladas.

Para terminar, se saca de la horma, se limpia y repasa el zapato, se coloca el pernito de cartón y finalmente se envasa en su caja con el papel y las etiquetas correspondientes.

Ruta: 1 ⇒ 2 ⇒ 4 ⇒ 5 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 16 ⇒ 22 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 32 ⇒ 33 ⇒ 35

El producto 5 es una sandalia tipo bio, similar a las clásicas Birkenstock, con tiras ajustables y una suela anatómica que ya lleva incorporada la plantilla. En este modelo, la piel va directamente en contacto con el pie, ya que no necesita forro ni cosidos: las tiras se presan directamente con la suela durante el montaje.

Primero se cortan las piezas de piel, se figuran y se colocan los ojetes metálicos junto con las hebillas en las tiras. Luego se pasa al repasado para comprobar que todo esté correcto antes de continuar.

En el ensamblaje, se monta el enfranque, se lija la piel y se prepara la suela, que se pega con la máquina de prensa. Después se introduce en el horno de baja temperatura para fijar bien las tiras y asegurar la unión.

Por último, se saca de la horma, se limpia, se coloca el pernito de cartón y se envasa en su caja con el papel y las etiquetas correspondientes.

Ruta: 1 ⇒ 2 ⇒ 17 ⇒ 16 ⇒ 22 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 32 ⇒ 33 ⇒ 35

El producto 6 es un zueco de estilo bio, cerrado por la parte delantera. En el cortado se preparan las piezas de piel que formarán el empeine y se hacen unos pequeños cortes por donde después pasará la tira con la hebilla. También se colocan los ojetes metálicos y la hebilla. Seguidamente se pasa al repasado, asegurándose de que las piezas queden bien terminadas antes del montaje.

En el ensamblaje, se ajusta la punta con la máquina correspondiente, dándole forma a la parte delantera del zueco, y se continúa con el montado del enfranque. A continuación, se lija la piel, se prepara la suela y se pega con la máquina de prensa. Luego se introduce en el horno de baja temperatura para asegurar la fijación completa de las tiras y el empeine.

Por último, se saca de la horma, se limpia bien el calzado, se coloca el pernito de cartón interno para que mantenga su forma y se envasa en su caja con las etiquetas y el papel pertinentes.

Ruta: 1 ⇒ 2 ⇒ 17 ⇒ 16 ⇒ 21 ⇒ 22 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 32 ⇒ 33 ⇒ 35

El producto 7 es una cuña con suela de esparto trenzada, muy típica del calzado de verano. En el cortado se preparan las piezas de piel necesarias, y después se pasa al aparado, donde se pegan los forros y se cosen las distintas partes del corte. Una vez repasado, se moldea la zona de la punta para darle forma y se pasa por el horno caliente antes del montado.

En el montaje se ajusta la punta y se coloca el enfranque, uniendo bien la piel con la estructura del zapato. Después se lija, se prepara la suela de esparto y se pega con la máquina de prensa. Para asegurar la unión, se introduce en el horno de baja temperatura.

En la parte final del proceso, se saca la horma, se pega la plantilla, se limpia y repasa bien el calzado. Por último, se coloca el pernito de cartón interno y se envasa en su caja con su papel y etiquetas.

Ruta: 1 ⇒ 8 ⇒ 9 ⇒ 16 ⇒ 18 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 30 ⇒ 32 ⇒ 33 ⇒ 35

El producto 8 es una sandalia de tacón. A diferencia del tacón cerrado, este modelo está formado por tiras de piel y PVC que sujetan el pie, lo que le da un aspecto más ligero y elegante.

Para fabricarla, primero se cortan y figuran las piezas de piel. Luego se rebajan y refuerzan las zonas necesarias, y se doblan los bordes. En el aparado, se pegan los forros y se cosen las distintas tiras y partes del zapato, que después pasan por el repasado.

A continuación, el corte pasa al horno caliente. Más tarde se lija la piel, se prepara y pega la suela con la prensa, y se vuelve a calentar en el horno de baja temperatura para fijar bien todo el conjunto.

En la fase final, se saca la horma y se atornilla el tacón. Luego se pega la plantilla y, si procede, se coloca una plantilla interior adicional. Se limpia y repasa el zapato, y se coloca el pernito de cartón para conservar la forma. Finalmente, se envasa con su papel y caja correspondiente, junto con las etiquetas.

Ruta: 1 ⇒ 2 ⇒ 4 ⇒ 5 ⇒ 6 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 16 ⇒ 20 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 29 ⇒ 30 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 35

El producto 9 es un botín estilo cowboy, propio de la temporada otoño-invierno. Este tipo de calzado lleva un proceso de fabricación más laborioso, ya que combina múltiples piezas de piel y acabados característicos.

Para su elaboración, primero se cortan las piezas de piel, se figuran y se prepara el acolchado. Después, se rebajan los bordes, se colocan los refuerzos y se doblan para conseguir un mejor acabado. En el aparado, se realiza el entreforrado y se pegan los forros, se cosen las distintas piezas que forman el cuerpo del botín y se añaden los respuntes decorativos. También se repasan las costuras para asegurar su aguante.

A continuación, el corte pasa por moldeado, donde se colocan los componentes metálicos y se moldean el tope y el contrafuerte.

En el ensamblaje, se calienta en el horno y se monta sobre la horma, ajustando punta, enfranque y talón. Después, se lija la superficie, se prepara y pega la suela con la prensa, y se vuelve a calentar en el horno de baja temperatura para asegurar la unión. Luego se pone la plantilla y se limpia y revisa el botín.

Finalmente, se introduce el pernito de cartón para mantener la forma y se embala junto con su caja y etiquetado.

Ruta: 1 ⇒ 2 ⇒ 3 ⇒ 4 ⇒ 5 ⇒ 6 ⇒ 7 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 11 ⇒ 12 ⇒ 13 ⇒ 15 ⇒ 16 ⇒ 17 ⇒ 18 ⇒ 19 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 23 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 35

El producto 10 es una bota de caña alta. Para su fabricación, primero se cortan las piezas de piel y se figuran. Después se rebajan los bordes, se ponen los refuerzos y se doblan las piezas para mejorar el acabado. En el aparado, que es la parte más completa del proceso, se realiza el entreforrado y se pegan los forros, se cosen las distintas piezas formando el cuerpo de la bota y se refuerzan las zonas del talón y la caña. A continuación, se cosen las

cremalleras. Luego se repasan todas las costuras para asegurar su resistencia y buen acabado.

Después, el corte pasa por moldeado, donde se incorporan los adornos metálicos, se forman el tope y el contrafuerte, y se ajusta la estructura de la caña para darle su forma característica.

En el ensamblaje, se calienta en el horno y se monta sobre la horma, ajustando punta, enfranque y talón. Luego se lija la bota, y se prepara y se pega la suela, que incluye el tacón, con la prensa, y se vuelve a calentar en el horno de baja temperatura para asegurar la unión.

Por último, se saca la horma, se coloca la plantilla, se limpia y revisa la bota, y se introduce el pernito de cartón para mantener su forma antes de embalarla con su caja y etiquetado.

Ruta: 1 ⇒ 2 ⇒ 4 ⇒ 5 ⇒ 6 ⇒ 7 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 11 ⇒ 13 ⇒ 14 ⇒ 15 ⇒ 16 ⇒ 17 ⇒ 18 ⇒ 19 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 23 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 35

El producto 11 es un deportivo técnico, distinto a los de piel tradicionales. En este caso, las piezas se cortan y figuran a partir de materiales textiles, a los que también se les colocan los refuerzos necesarios para aportar estructura. A continuación, se realiza el entreforrado y se comienzan a coser las distintas partes del corte. Después se pegan los forros, se unen los talones, se cosen más forros y las lengüetas para completar la parte superior del deportivo.

En el montado, se ajusta la punta sobre la horma para dar forma al zapato, y seguidamente se prepara y se pega la suela mediante la máquina de prensa. Luego pasa por el horno de baja temperatura para asegurar la adhesión. Finalmente, se saca la horma, se coloca la

plantilla, se revisa y limpia el calzado, se ponen los cordones y se introduce el pernito de cartón antes de envasarlo con su caja y etiquetado.

Ruta: 1 ⇒ 2 ⇒ 5 ⇒ 7 ⇒ 9 ⇒ 8 ⇒ 15 ⇒ 10 ⇒ 14 ⇒ 21 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 34 ⇒ 35

El producto 12 es un deportivo estilo Gazelle, un modelo clásico con un diseño más sencillo y menos recargado que otros deportivos. Para su fabricación, se cortan las piezas de piel, se figuran y se colocan los refuerzos para el aparado. En esta fase, se prepara el entreforrado, se pegan los forros y se cosen las distintas partes del corte, uniendo las piezas principales del zapato. Después se cosen los talones y se realiza un repaso general para asegurar un buen rematado.

En la zona de moldeado, se forman el tope y el contrafuerte. En el ensamblaje se pasa por el horno caliente para ablandar la piel y facilitar el montado de la punta, el enfranque y el talón. Una vez montado, se lija la piel, se prepara la suela y se pega con la máquina de prensa. Después se introduce en el horno de baja temperatura para reforzar la unión entre las piezas.

Por último, se saca la horma, se coloca la plantilla y se revisa el acabado del deportivo. Se limpian y ajustan los detalles finales, se introduce el pernito de cartón, se ponen los cordones y finalmente se envasa en su caja con su etiquetado conveniente.

Ruta: 1 ⇒ 2 ⇒ 5 ⇒ 7 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 11 ⇒ 15 ⇒ 16 ⇒ 18 ⇒ 19 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 23 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 34 ⇒ 35

Para fabricar el producto 13, un botín negro se empieza cortando las pieles y forros, se preparan las piezas en el figurado y se divide el acolchado. Se rebaja la piel y se ponen los refuerzos internos. Después, se pasa al aparado, donde empieza la parte más detallada del proceso. Primero, se realiza el entreforrado y se pegan los forros. Se cosen las distintas

piezas del corte con la máquina plana, uniendo las capas de piel superpuestas que dan forma al diseño. Luego, se utiliza la máquina automática lateral para cerrar el zapato, se cose todo el corte y más adelante la máquina de columna para zonas más estrechas o curvas, como la parte de las lenguas o el talón. Para finalizar, lo revisan las repasadoras.

A continuación, se colocan todos los detalles metálicos y el corte se moldea formando el tope y el contrafuerte, pasando por el horno caliente antes del montado. Sobre la horma, se ajustan la punta, el enfranque y el talón para dar la forma definitiva al botín. En la zona de ensamblaje, se lija la parte inferior del zapato, se prepara la suela y se fija con la prensa. Luego se pasa por el horno de baja temperatura para reforzarlo.

Finalmente, se retira la horma, se coloca la plantilla, y se hace una limpieza final. Después se añaden los cordones y el pernito de cartón para mantener la forma, y el botín se prepara para su embalaje en la caja.

Ruta: 1 ⇒ 2 ⇒ 3 ⇒ 4 ⇒ 7 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 12 ⇒ 13 ⇒ 14 ⇒ 15 ⇒ 16 ⇒ 17 ⇒ 18 ⇒ 19 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 23 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 34 ⇒ 35

El producto 14 es un deportivo con cremallera lateral y cordones, que combina diferentes piezas de piel con detalles decorativos. Para fabricarlo, primero se cortan las pieles y se figuran. Después, se prepara el acolchado, se rebajan los bordes, y se doblan las piezas ya reforzadas para mejorar el acabado.

En el aparado, se introducen los refuerzos entre piel y forro, se pegan los forros y se cosen las distintas piezas del corte. Este modelo pasa por varias máquinas de coser, ya que tiene muchas uniones y piezas superpuestas: primero se cosen las partes laterales, luego se realiza el cosido de todo el corte y, a continuación, las zonas de la lengüeta y el talón. También se repasan las costuras para asegurar la solidez.

A continuación, pasa por la zona de moldeado, donde se colocan los ojetes metálicos, se forma el tope y el contrafuerte, y se calienta en el horno. Luego, se monta el zapato sobre la horma, ajustando la punta, el enfranque y el talón.

En el ensamblaje, se lija la piel. Después se prepara y pega la suela, que incorpora una ligera plataforma, utilizando la prensa. Finalmente, se vuelve a calentar en el horno de baja temperatura para sellar bien las dos partes.

Finalmente, en el acabado, se saca la horma, se coloca la plantilla y se limpia el zapato. Después, se introduce el pernito de cartón para mantener la forma, se añaden los cordones y se embala en su caja con las etiquetas oportunas.

Ruta: 1 ⇒ 2 ⇒ 3 ⇒ 4 ⇒ 7 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 11 ⇒ 12 ⇒ 13 ⇒ 14 ⇒ 15 ⇒ 16 ⇒ 17 ⇒ 18 ⇒ 19 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 23 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 34 ⇒ 35

El producto 15 es un deportivo con combinación de colores. Para su fabricación, primero se realiza el corte de las piezas en las distintas tonalidades y materiales, asegurando que cada una encaje con precisión en el diseño final. Luego se figuran las pieles, se divide el acolchado interior y se rebajan los bordes.

En la fase de aparado, se insertan los refuerzos interiores entre piel y forro, y se ensamblan las piezas mediante varias costuras, utilizando diferentes máquinas según la forma y grosor de cada parte. Se unen las piezas laterales, la zona del talón y la puntera, integrando los distintos colores. Después, se cosen las lengüetas y se revisan las uniones antes de continuar con el montaje.

En el moldeado, se colocan los ojetes metálicos, se forman el tope y el contrafuerte, y se calienta el conjunto para que tome su forma definitiva. Luego, se pasa al ensamblaje, donde sobre la horma, se ajustan punta, enfranque y talón, asegurando que todo quede bien alineado. Posteriormente, se lija el borde inferior, se prepara la suela y se adhiere

con la prensa. Tras esto, se introduce en el horno de baja temperatura para consolidar el pegado.

Por último, en la fase de acabado, se retira la horma, se coloca la plantilla y se realiza una limpieza general. Se añaden los cordones, se inserta el pernito de cartón para conservar la forma y se envasa cuidadosamente con su caja y etiquetas.

Ruta: 1 ⇒ 2 ⇒ 3 ⇒ 4 ⇒ 7 ⇒ 8 ⇒ 9 ⇒ 10 ⇒ 11 ⇒ 12 ⇒ 13 ⇒ 14 ⇒ 15 ⇒ 16 ⇒ 17 ⇒ 18 ⇒ 19 ⇒ 20 ⇒ 21 ⇒ 22 ⇒ 23 ⇒ 24 ⇒ 25 ⇒ 26 ⇒ 27 ⇒ 28 ⇒ 31 ⇒ 32 ⇒ 33 ⇒ 34 ⇒ 35

### 3.5 ORGANIZACIÓN Y RECURSOS HUMANOS

La fábrica cuenta con personal distribuido por zonas según las etapas del proceso. En cada una trabajan los operarios que se encargan directamente de las tareas que corresponden, ya sea el manejo de máquinas, el aparado de las piezas o el montaje final del calzado. Cada operario conoce bien su puesto y las operaciones que realiza, y suele mantenerse en la misma sección durante toda la jornada para mantener la continuidad del trabajo.

La coordinación entre zonas es bastante lineal. Cuando se termina el último proceso de una etapa, el producto pasa directamente a la siguiente, siguiendo el orden establecido. Cada par cuenta con la información necesaria para identificar su modelo y conocer cuál es el siguiente proceso que debe realizarse. Esto permite que el flujo entre zonas sea continuo y no sea necesario ningún control intermedio más allá de la propia organización del trabajo.

El trabajo se organiza en turnos de lunes a viernes, con jornadas de ocho horas. Durante la temporada de producción se completa todo el volumen encargado, ajustando el ritmo de trabajo según la carga de pedidos.

Primero está la zona de cortado, donde se preparan las piezas que formarán el zapato.

Etapa 1 (Corte): cuenta con 6 máquinas de corte automático y troquelado, donde se cortan las piezas de piel o material sintético según los moldes de cada modelo.

Etapa 2 (Figurado): dispone de 3 operarios que perfilan los bordes de las piezas ya cortadas.

Etapa 3 (Dividido y acolchado): con 3 máquinas, se ajusta el grosor de las piezas y se aplica el acolchado cuando el diseño lo requiere.

Etapa 4 (Rebajado): 2 máquinas afinan los bordes para facilitar el posterior montaje.

Etapa 5 (Colocación de refuerzos): 1 operario prepara las zonas del tope y el contrafuerte, colocando los refuerzos necesarios.

Etapa 6 (Doblado): 1 máquina dobla los bordes para mejorar el acabado exterior de las piezas.

La zona de apurado es la parte más laboriosa del proceso, donde se cosen y ensamblan las piezas que darán forma al corte.

Etapa 7 (Entreforrado): 1 operario fija los refuerzos y une el forro con el corte.

Etapa 8 (Coser o pegar forros): 2 máquinas aplican adhesivos o cosen los forros al conjunto, según el tipo de material.

Etapas 9, 10 y 11 (Cosido de piel): cada una cuenta con 2 máquinas de cosido lineal que unen las distintas partes del zapato.

Etapa 12 (Máquina automática lateral): 1 máquina une los laterales del corte con precisión.

Etapa 13 (Cosido completo): 2 máquinas completan la unión de todas las piezas del corte.

Etapa 14 (Coser lenguas): 2 máquinas cosen las lenguas al resto del corte.

Etapa 15 (Coser talones): 2 máquinas cierran el talón y refuerzan la unión trasera.

Etapa 16 (Repaso): 3 operarias revisan el corte completo, eliminan hilos sueltos y preparan las piezas para el montaje.

En la zona de moldeado se preparan los componentes estructurales del zapato.

Etapa 17 (Colocación de ojetes metálicos y/o hebillas): 1 prensa remachadora coloca los componentes metálicos necesarios.

Etapa 18 (Moldeado del tope): 2 máquinas moldean la parte frontal del zapato.

Etapa 19 (Moldeado del contrafuerte): 2 máquinas dan forma a la zona del talón para garantizar la sujeción.

En la zona de ensamblaje comienza el proceso de unión entre el corte y la suela.

Etapa 20 (Horno caliente): un horno con capacidad para 100 pares, controlado por operarios, calienta las piezas para facilitar el montaje.

Etapa 21 (Montado de punta): 1 máquina, manejada por un operario especializado, ajusta la punta del zapato sobre la horma.

Etapa 22 (Montado de enfranque): 1 máquina trabaja la parte central del zapato, asegurando la forma.

Etapa 23 (Montado de talón): 1 operario monta el talón con la máquina correspondiente.

Etapa 24 (Lijado de piel): 2 operarios trabajan con una máquina de rodillos para lijar y preparar la superficie.

Etapa 25 (Preparado de suela): 2 operarios preparan las suelas antes del pegado, aplicando los productos necesarios.

Etapa 26 (Pegado de suela): 2 operarios fijan la suela al corte con una prensa.

Etapa 27 (Horno de baja temperatura): un horno de baja temperatura con capacidad para 100 pares activa el adhesivo y fija la unión.

Por último, la zona de acabado y envasado completa el proceso y deja el producto listo para su salida.

Etapa 28 (Sacar horma): 2 operarios extraen la horma de cada zapato con una máquina específica.

Etapa 29 (Atornillar tacón): 1 operario coloca los tacones con una máquina de clavar o atornillar.

Etapa 30 (Pegar plantilla): 1 máquina aplica adhesivo y fija la plantilla en su posición.

Etapa 31 (Colocación de plantilla interior): 2 operarios colocan las plantillas y revisan el acabado interno.

Etapa 32 (Repaso y limpieza): 3 operarios se encargan del control de calidad, limpieza y verificación final.

Etapa 33 (Pernito interno): 2 operarios insertan el pernito o molde interno para mantener la forma.

Etapa 34 (Poner cordones): 2 operarios colocan los cordones según el modelo.

Etapa 35 (Envasado): 2 operarios se encargan de embalar los pares en sus cajas con papel, separadores y etiquetas, dejándolos listos para el envío o almacenamiento.

En conjunto, la planta reúne una amplia variedad de oficios que dan forma al proceso completo de fabricación. Desde los cortadores y dobladores que preparan las piezas iniciales, hasta las aparadoras y repasadoras que unen y perfeccionan cada detalle, todos los puestos tienen un papel esencial. También están los moldeadores, los operarios de montaje, los encargados del mantenimiento de hornos y prensas, y el personal de acabado y envasado, que trabaja con cuidado para que cada par quede listo para su entrega. El trabajo de todos ellos, coordinado día a día, es el que realmente hace posible que la producción avance y que cada zapato llegue a su destino en perfectas condiciones.

Una vez descrita la estructura de la fábrica y la forma en que se trabaja en cada etapa, el siguiente paso es centrarse en cómo se puede planificar y optimizar todo ese proceso mediante modelos matemáticos.

## 4. MODELOS MATEMÁTICOS Y DESARROLLO DEL PROYECTO

### 4.1 INTRODUCCIÓN AL MODELADO

Este apartado del trabajo se va a centrar en los modelos matemáticos desarrollados para optimizar la planificación de la producción. El propósito es encontrar la ruta más eficiente para los productos, asignándolos a las máquinas de forma que el tiempo total de producción sea el menor posible.

A lo largo del desarrollo se han ido probando distintos enfoques para mejorar la planificación. En primer lugar, se describe un modelo matemático que trabaja de forma par a par, donde el modelo genera soluciones mezclando continuamente pares de distintos tipos de productos. Después, se presenta un segundo modelo que trabaja por bloques de producto: parte de la demanda completa y va asignando los pares a las máquinas disponibles en cada etapa. En un tercer intento, se mantiene el mismo modelo anterior, pero simplificando el sistema al suponer una única máquina con mayor velocidad de producción dependiendo del número de máquinas inicial por etapa, con el objetivo de facilitar los cálculos y reducir la carga computacional. Por último, se desarrolla una heurística basada en órdenes de producción y permutaciones, que permite encontrar soluciones muy cercanas al óptimo en mucho menos tiempo.

Cada modelo ha permitido analizar distintas formas de plantear el problema, se han corregido los problemas que iban apareciendo y poco a poco nos acercábamos a una planificación más eficiente y coherente con el funcionamiento real de la planta.

### 4.2 PRIMER MODELO MATEMÁTICO: ENFOQUE PAR A PAR

El primer modelo matemático que se ha desarrollado busca optimizar la planificación de la producción trabajando con el calzado par a par, es decir, en lugar de tratar la demanda como un bloque global, el modelo considera cada unidad como un par de zapatos independiente. A partir de toda la demanda existente, decide el momento en que entra cada par, a la etapa y máquina correspondiente, para encontrar una ruta que reduzca al máximo el tiempo total de fabricación. Este tiempo total se conoce como *makespan*, y

representa el instante en el que termina la última operación del sistema (Hazewinkel, 2021).

El modelo matemático tiene una serie de conjuntos, variables y parámetros que hacen que se puedan crear las restricciones reproduciendo una planta de producción.

En el modelo se definen varios conjuntos que ayudan a organizar los elementos que intervienen en la producción:

$J$ : es el conjunto de *productos* con los que trabaja el sistema. Cada producto se identifica con un índice  $j$ . Sirve para que el modelo sepa cuántos tipos de producto hay y pueda distinguirlos entre sí.

$K$ : es el conjunto de *etapas* del proceso productivo, indexado por  $k$ . Cada etapa representa una parte distinta de la fabricación (por ejemplo, corte, moldeado, horno, etc.).

$M_k$ : representa el conjunto de *máquinas* disponibles en la etapa  $k$ , indexadas por  $m$ . No todas las etapas tienen el mismo número de máquinas, así que este conjunto depende de la etapa.

$U_j$ : es el conjunto de *unidades* del producto  $j$ , indexadas por  $u$ . Cada unidad corresponde a un par de zapatos, y el número total de unidades de cada producto viene dado por  $|U_j| = d_j$ , donde  $d_j$  es la demanda del producto  $j$ .

Los parámetros son los valores fijos que el modelo necesita para poder funcionar y tomar decisiones. No cambian durante la ejecución, sino que sirven como datos de entrada.

$d_j$ : representa la *demanda* del producto  $j$ , es decir, cuántas unidades de ese producto se deben fabricar. Cada unidad se corresponde con un par de zapatos en el modelo.

$r_{jk}$ : indica el *tiempo de proceso* (en minutos por unidad) que tarda una unidad del producto  $j$  en completarse en la etapa  $k$ . Ese tiempo no depende de la máquina concreta, solo de la etapa.

$M$ : es una *constante grande* (conocida como “Big-M”) que se utiliza en algunas restricciones del modelo para permitir o desactivar condiciones según sea necesario. No

tiene un valor físico real, simplemente ayuda al modelo a activar o desactivar ciertas condiciones cuando hace falta.

$T_{hor}$ : representa el horizonte temporal máximo, es decir, el tiempo total disponible para terminar toda la producción.

Las variables son los valores que el modelo decide para obtener la mejor planificación posible. Representan los tiempos, asignaciones y secuencias de producción que se determinarán al resolver el modelo.

$S_{juk} \geq 0$ : indica el *tiempo de inicio* de la unidad  $u$  del producto  $j$  en la etapa  $k$ . Es el momento en el que esa unidad comienza a procesarse en la etapa correspondiente.

$C_{juk} \geq 0$ : representa el *tiempo de finalización* de la unidad  $u$  del producto  $j$  en la etapa  $k$ . Se calcula a partir del tiempo de inicio y del tiempo de proceso.

$z_{jukm} \in \{0,1\}$ : variable binaria que toma el valor 1 si la unidad  $u$  del producto  $j$  se procesa en la *máquina*  $m$  perteneciente a la etapa  $k$ , y 0 en caso contrario. Permite al modelo decidir en qué máquina concreta se procesará cada unidad.

$y_{(ju),(j'u'),k,m} \in \{0,1\}$ : variable binaria que toma el valor 1 si la unidad  $(j, u)$  se procesa *antes que*  $(j', u')$  en la máquina  $m$  de la etapa  $k$ . Se define solo para los pares de unidades que comparten la misma máquina, y permite establecer el orden de procesamiento dentro de cada máquina.

$C_{max} \geq 0$ : representa el *makespan*, es decir, el tiempo total necesario hasta que se termina la última unidad del sistema. Es la variable objetivo del modelo.

La función objetivo busca minimizar  $C_{max}$ , que es el tiempo total que tarda en completarse toda la producción. En otras palabras, el modelo intenta que el proceso termine lo antes posible, ajustando los tiempos de inicio, las máquinas y el orden de los productos para que el conjunto de unidades acabe en el menor tiempo posible.

$$\min C_{max}$$

La restricción 1 controla la duración de las operaciones de cada unidad:

$$C_{juk} = S_{juk} + r_{jk} \quad \forall j \in J, \forall u \in U_j, \forall k \in K \quad (1)$$

Esta restricción indica que el tiempo de fin de una unidad en una máquina es igual al tiempo de inicio en dicha máquina más el tiempo necesario para completarse, garantizando que el modelo respete la duración real del proceso.

La restricción 2 corresponde a la de asignación única de máquina (por unidad).

$$\sum_{m \in M_k} z_{jukm} = 1 \quad \forall j \in J, \forall u \in U_j, \forall k \in K \quad (2)$$

Esta restricción se asegura de que cada unidad solo se asigne a una máquina dentro de cada etapa. En otras palabras, no se puede fabricar el mismo par de zapatos en dos máquinas distintas dentro de la misma etapa.

La restricción 3 es la de precedencia entre etapas consecutivas (por unidad).

$$S_{ju,k+1} \geq C_{juk} \quad \forall j \in J, \forall u \in U_j, \forall k = 1, \dots, |K| - 1 \quad (3)$$

Esta restricción se encarga de que una unidad no pueda empezar la siguiente etapa antes de haber terminado la anterior. Es decir, mantiene el orden lógico del proceso: primero se acaba una etapa y luego se pasa a la siguiente, evitando que el modelo salte pasos o solape tareas que deben ir una detrás de otra.

Las restricciones 4 y 5 regulan el no solapamiento en la misma máquina (activado solo si ambas asignaciones existen).

$$S_{j'u'k} \geq C_{juk} - M(1 - y_{(ju),(j'u'),k,m}) - M(2 - z_{jukm} - z_{j'u'km}) \quad (4)$$

$$S_{juk} \geq C_{j'u'k} - M y_{(ju),(j'u'),k,m} - M (2 - z_{jukm} - z_{j'u'km}) \quad (5)$$

Esta restricción evita que dos unidades se procesen al mismo tiempo en la misma máquina. Solo se activa si ambas unidades están asignadas a esa máquina. La variable binaria  $y_{(ju),(j'u'),k,m}$  indica cuál de las dos unidades se procesa primero. Así se garantiza que no haya solapamientos imposibles y que cada máquina procese una unidad a la vez.

La restricción 6 es la de cotas del horizonte temporal.

$$0 \leq S_{juk} \leq T_{hor}, \quad 0 \leq C_{juk} \leq T_{hor} \quad \forall j \in J, \forall u \in U_j, \forall k \in K \quad (6)$$

Esta restricción se encarga de mantener todos los tiempos de inicio y fin dentro del horizonte temporal definido por el modelo. Asegura que ninguna operación comience antes del tiempo 0 ni termine después del límite máximo  $T_{hor}$ . Sirve para que todas las operaciones se mantengan dentro del tiempo previsto y no se salgan del horario de planificación.

La restricción 7 es la definición del *makespan*.

$$C_{max} \geq C_{juk} \quad \forall j \in J, \forall u \in U_j, \forall k \in K \quad (7)$$

Esta restricción indica que el  $C_{max}$  debe ser al menos igual al mayor de todos los tiempos de finalización. De esta manera, representa el momento en que termina el último par en pasar por todas sus etapas.

En resumen, el modelo matemático consiste en minimizar  $C_{max}$  sujeto a las restricciones (1)–(7) y a la definición de variables.

### **Función objetivo**

$$\min C_{max} \quad (\text{FO})$$

## Restricciones

(R1) Duración de las operaciones (por unidad)

$$C_{juk} = S_{juk} + r_{jk} \quad \forall j \in J, \forall u \in U_j, \forall k \in K \quad (1)$$

(R2) Asignación única de máquina (por unidad)

$$\sum_{m \in M_k} z_{jukm} = 1 \quad \forall j \in J, \forall u \in U_j, \forall k \in K \quad (2)$$

(R3) Precedencia entre etapas consecutivas (por unidad)

$$S_{ju,k+1} \geq C_{juk} \quad \forall j \in J, \forall u \in U_j, \forall k = 1, \dots, |K| - 1 \quad (3)$$

(R4-5) No solapamiento en la misma máquina (solo si ambas asignaciones existen)

$$S_{j'u'k} \geq C_{juk} - M (1 - y_{(ju),(j'u'),k,m}) - M (2 - z_{jukm} - z_{j'u'km}) \quad (4)$$

$$S_{juk} \geq C_{j'u'k} - M y_{(ju),(j'u'),k,m} - M (2 - z_{jukm} - z_{j'u'km}) \quad (5)$$

(R6) Cotas del horizonte temporal

$$0 \leq S_{juk} \leq T_{hor}, \quad 0 \leq C_{juk} \leq T_{hor} \quad \forall j \in J, \forall u \in U_j, \forall k \in K \quad (6)$$

(R7) Definición del makespan

$$C_{max} \geq C_{juk} \quad \forall j \in J, \forall u \in U_j, \forall k \in K \quad (7)$$

Nota: Todas las variables están definidas según se indica en la sección de conjuntos, parámetros y variables.

A partir de este modelo matemático se programó el código correspondiente en Python. El funcionamiento era correcto, así que se realizaron pruebas iniciales con dos productos y

dos etapas. Conforme se aumentaba la demanda, el modelo era computacionalmente más lento, ya que no escalaba bien.

El máximo de demanda que permitió obtener una solución óptima fue de 4 pares por producto, es decir, 8 unidades en total. Al intentar resolverlo con 5 pares por producto, el tiempo de ejecución creció de forma exponencial y no se consiguió una solución óptima. En ese caso, la ejecución se detuvo tras 54.440 segundos, más de 15 horas.

A continuación, se muestran en la Tabla 1 los resultados obtenidos.

**Tabla 1**

*Resultados del primer modelo matemático*

N.º de productos	N.º de etapas	N.º pares totales	Tiempo ejecución (s)	Makespan (min)	¿Solución óptima?
2	2	2	0,05	3,20	Sí
2	2	4	0,02	3,20	Sí
2	2	6	0,03	3,20	Sí
2	2	8	0,40	4,40	Sí
2	2	10	+54440		No

Con esto se puede ver que el modelo funciona bien cuando el problema es pequeño, pero a medida que se aumenta la cantidad de pares, el tiempo de cálculo crece exponencialmente. Aunque los resultados son correctos y óptimos, el modelo no llega a ser práctico para situaciones realistas de producción. Incluso con solo dos productos, dos etapas, seis máquinas en la primera y tres en la segunda, el número de variables ya es tan elevado que el solver de Gurobi tarda horas en obtener la solución óptima.

Dadas las limitaciones de este modelo, se decide buscar otras alternativas y desarrollar un nuevo modelo matemático que trabaje con menos variables. Este segundo modelo planifica los productos por bloques en lugar de par a par, eliminando así el subíndice  $u$ . Con esto se pretende manejar un mayor volumen de producción y obtener soluciones en tiempos más razonables, de manera que puedan usarse en una planta de fabricación de calzado.

### 4.3 SEGUNDO MODELO MATEMÁTICO: POR BLOQUES DE PRODUCTOS

El modelo matemático que se desarrolla ahora pretende ser menos costoso a nivel de carga computacional. Lo que cambia respecto al anterior es que este ya no trabaja par a par, sino que considera la demanda de cada tipo de producto como un bloque que va avanzando por las distintas etapas, y que se va dividiendo entre todas las máquinas de esa etapa.

De esta manera, el modelo tiene que manejar muchas menos variables, lo que reduce el tiempo de ejecución del programa y puede obtener soluciones en un menor tiempo.

En este modelo se mantienen algunas de las restricciones del anterior, pero adaptadas al nuevo planteamiento, y se añaden otras nuevas para que se consiga representar la planta. El objetivo sigue siendo planificar la producción de manera optimizada para poder fabricar en el menor tiempo posible.

Al igual que el primer modelo matemático, este también consta de una serie de conjuntos, parámetros y variables.

Los conjuntos son únicamente estos tres:

$J$ : es el conjunto de productos que se fabrican, y cada uno se identifica con el índice  $j$ .

$K$ : representa las etapas del proceso productivo, como pueden ser corte, pegado o acabado, y se identifica con el índice  $k$ .

$M_k$ : es el conjunto de máquinas que hay en cada etapa  $k$ , y cada máquina se representa con el índice  $m$ .

Los parámetros son los datos constantes que el modelo necesita para resolver el problema. Se utilizan como información de entrada y no cambian durante la ejecución.

$d_j$ : demanda del producto  $j$ , en unidades.

$p_{kj} \in \{0,1\}$ : toma el valor 1 si el producto  $j$  pasa por la etapa  $k$ ; en caso contrario, vale 0. Permite que el modelo sepa qué etapas necesita cada producto.

$T_{km}$ : representa el tiempo disponible en la máquina  $m$  de la etapa  $k$ . Se utiliza para que la asignación no supere la capacidad de cada máquina.

$r_{jkm}$ : tiempo de proceso por unidad del producto  $j$  en la máquina  $m$  de la etapa  $k$ . En este modelo el tiempo depende también de la máquina, lo que permite reflejar diferencias de rendimiento entre ellas.

$U$ : constante grande (Big-M) usada para activar o desactivar condiciones dentro de algunas restricciones.

$\varepsilon > 0$ : cantidad mínima procesable. Sirve para evitar que el modelo asigne cantidades nulas cuando se distribuye la carga entre máquinas.

En este modelo, las variables son los valores que el programa tiene que calcular para planificar la producción de manera óptima, pero considerando bloques en lugar de par a par.

$q_{jkm} \geq 0$ : representa la cantidad del producto  $j$  que se fabrica en la máquina  $m$  de la etapa  $k$ . Es decir, cuántas unidades de ese producto se asignan a cada máquina.

$z_{jkm} \in \{0,1\}$ : toma el valor 1 si la máquina  $m$  procesa alguna cantidad del producto  $j$  en la etapa  $k$ , y 0 si no lo hace.

$S_{jkm} \geq 0$ : es el momento en que empieza la producción del bloque del producto  $j$  en la máquina  $m$  de la etapa  $k$ .

$C_{jkm} \geq 0$ : indica el momento en que termina ese bloque de producción.

$y_{jj',km} \in \{0,1\}$ : sirve para establecer el orden de fabricación cuando dos productos diferentes usan la misma máquina. Si vale 1, significa que el producto  $j$  se fabrica antes que el  $j'$ .

$C_{max} \geq 0$ : simboliza el makespan, que es el tiempo total que tarda en completarse toda la producción. Es el valor que el modelo intenta minimizar.

La función objetivo de este modelo tiene el mismo propósito general: conseguir que toda la fabricación del calzado termine lo antes posible. En este caso, lo que se busca es minimizar el valor de  $C_{max}$ , que representa el momento en el que finaliza el último bloque de producto.

En resumen, el modelo intenta ajustar la asignación de cantidades, el uso de las máquinas y los tiempos de inicio y fin de cada bloque para que el proceso completo tarde lo mínimo posible.

$$\min C_{max}$$

La restricción 8 controla que se cumpla la demanda en la última etapa:

$$\sum_{m \in M_{k^{last}(j)}} q_{j k^{last}(j) m} = d_j \quad \forall j \in J \quad (8)$$

Aquí,  $k^{last}(j)$  representa la última etapa por la que pasa el producto  $j$ . Lo que hace la restricción es que, al llegar a la última etapa de cada producto, se sumen todas las unidades asignadas a las máquinas y coincidan con la demanda total. Así nos aseguramos de que no falte ningún par por producir y que se cumpla la cantidad que realmente se necesita.

La restricción 9 mantiene el equilibrio del flujo entre etapas sucesivas:

$$\sum_{m \in M_{k_1}} q_{j k_1 m} = \sum_{m \in M_{k_2}} q_{j k_2 m} \quad \forall j \in J, (k_1, k_2) \in Succ(K_j) \quad (9)$$

$Succ(K_j)$  representa el conjunto de pares de etapas consecutivas por las que pasa el producto  $j$ . Lo que busca la restricción es que la cantidad que sale de una etapa sea exactamente la misma que entra en la siguiente. De esta forma, no se pierden unidades por el camino y todo lo que se elabora en un proceso continúa su recorrido por la línea de fabricación.

La restricción 10 limita el trabajo de cada máquina en el tiempo que tiene asignado.

$$\sum_{j \in J} r_{jkm} q_{jkm} \leq T_{km} \quad \forall k \in K, m \in M_k \quad (10)$$

Esto significa que al sumar todos los tiempos que tardan los productos en una misma máquina, el resultado no puede ser mayor que el tiempo disponible para esa máquina. Así se evita que el modelo le asigne más carga de la que puede asumir.

La restricción 11 relaciona que el inicio y fin de todos los productos en una etapa:

$$C_{jkm} = \begin{cases} S_{jkm} + r_{jkm} q_{jkm}, & \text{si } k \notin \{20,27\}, \\ S_{jkm} + r_{jkm} z_{jkm}, & \text{si } k \in \{20,27\}, \end{cases} \quad \forall j \in J, k \in K, m \in M_k \quad (11)$$

Esta restricción marca la relación entre cuándo empieza y cuándo termina cada bloque. En la mayoría de las etapas, el tiempo de fin se calcula sumando al tiempo de inicio lo que tarda cada unidad por la cantidad que pasa por esa máquina. En cambio, en las etapas 20 y 27 (los hornos), el tiempo depende de los ciclos que se necesiten según la capacidad, así que en lugar de usar la cantidad procesada se activa con la variable  $z_{jkm}$ .

La restricción 12 asegura la precedencia entre etapas:

$$C_{jk_1m_1} \leq S_{jk_2m_2} \quad \forall j \in J, (k_1, k_2) \in Succ(K_j), m_1 \in M_{k_1}, m_2 \in M_{k_2} \quad (12)$$

Restringe que un producto no comience la siguiente etapa (su tiempo de inicio  $S$ ) hasta que haya terminado la etapa actual (su tiempo de fin  $C$ ). De esta manera, se mantiene el orden de las etapas y ninguna operación empieza antes de que la anterior haya finalizado.

Las restricciones 13 y 14 indican que no puede haber solapamiento en una misma máquina cuando dos productos se procesan en ella.

$$S_{j'km} \geq C_{jkm} - U(1 - y_{jj'km}) - U(2 - z_{jkm} - z_{j'km}) \quad (13)$$

$$S_{jkm} \geq C_{j'km} - Uy_{jj'km} - U(2 - z_{jkm} - z_{j'km}) \quad (14)$$

La parte  $-U(2 - z_{jkm} - z_{j'km})$  hace que la restricción deje de actuar cuando alguno de los dos productos no está asignado a esa máquina.

Esta restricción sirve para evitar que dos productos se procesen en la misma máquina al mismo tiempo. Es decir, si una máquina ya está ocupada, el siguiente producto tiene que esperar a que el anterior termine para poder procesarse. El modelo compara los tiempos de inicio y fin, además del orden entre productos y si se tienen que procesar en esa etapa.

La restricción 15 conecta la cantidad procesada con la activación de la máquina.

$$\varepsilon z_{jkm} \leq q_{jkm} \leq U z_{jkm} \quad \forall j, k, m \quad (15)$$

Es decir, si la máquina no se usa ( $z_{jkm} = 0$ ), no puede producir nada. Y si está activa ( $z_{jkm} = 1$ ), entonces puede fabricar una cierta cantidad dentro de los límites. Así se evita que el modelo invente producción en máquinas que realmente no están funcionando.

Y además, se añade la siguiente condición:

$$z_{jkm} \leq p_{kj} \quad \forall j, k, m \quad (16)$$

Esta condición dicta que la máquina solo pueda activarse si la etapa forma parte de la ruta del producto.

Por último, dentro de esta restricción:

$$q_{jkm} \leq 1000 z_{jkm} \quad \forall j \in J, k \in \{20,27\}, m \in M_k \quad (17)$$

En las etapas de los hornos (20 y 27) se limita la cantidad para evitar que el modelo cargue más de la capacidad permitida por el horno en un solo ciclo.

La restricción 18 regula el orden de activación entre productos cuando comparten una misma máquina:

$$y_{jj',km} \leq z_{jkm}, \quad y_{jj',km} \leq z_{j'km} \quad \forall j \neq j', k, m \quad (18)$$

Esta restricción evita al modelo cálculos innecesarios y así ahorra tiempo de ejecución. Esto se debe a que solo activa la variable de orden  $y_{jj',km}$  si dos productos están asignados a la misma máquina de esa etapa; de lo contrario, no se activa.

Para terminar con las restricciones, la restricción 19 define el makespan:

$$C_{\max} \geq C_{jkm} \quad \forall j, k, m \quad (19)$$

Su función es marcar el punto final de toda la planificación.  $C_{\max}$  recoge el instante en el que termina el último producto, así que su valor debe ser igual o mayor que cualquiera de los tiempos de fin. El modelo se entera así de cuándo acaba todo el proceso y con esto puede reducir ese tiempo total al mínimo posible.

Por tanto, el objetivo del modelo es minimizar  $C_{\max}$  teniendo en cuenta las restricciones (8)-(19) y las variables que reúnen.

### **Función objetivo**

$$\min C_{\max} \quad (\text{FO})$$

### **Restricciones**

(R8) Demanda en la última etapa

Sea  $k^{\text{last}}(j) = \max \{k \in K: p_{kj} = 1\}$ . Entonces

$$\sum_{m \in M_{k^{\text{last}}(j)}} q_{j k^{\text{last}}(j) m} = d_j \quad \forall j \in J \quad (8)$$

(R9) Flujo entre etapas (conservación agregada)

Para cada producto  $j$  y par sucesivo de etapas  $(k_1, k_2)$ .

$$\sum_{m \in M_{k_1}} q_{j k_1 m} = \sum_{m \in M_{k_2}} q_{j k_2 m} \quad \forall j \in J, (k_1, k_2) \in \text{Succ}(K_j) \quad (9)$$

(R10) Capacidad de máquina (tiempo total disponible)

$$\sum_{j \in J} r_{jkm} q_{jkm} \leq T_{km} \quad \forall k \in K, m \in M_k \quad (10)$$

(R11) Relación inicio-fin por bloque

$$C_{jkm} = \begin{cases} S_{jkm} + r_{jkm} q_{jkm}, & \text{si } k \notin \{20, 27\}, \\ S_{jkm} + r_{jkm} z_{jkm}, & \text{si } k \in \{20, 27\}, \end{cases} \quad \forall j \in J, k \in K, m \in M_k \quad (11)$$

En las etapas 20 y 27 (hornos), el tiempo de proceso depende del número de ciclos necesarios según la capacidad del horno, y se activa mediante  $z_{jkm}$  en lugar de  $q_{jkm}$ .

(R12) Precedencia entre etapas (temporal)

$$C_{j k_1 m_1} \leq S_{j k_2 m_2} \quad \forall j \in J, (k_1, k_2) \in \text{Succ}(K_j), m_1 \in M_{k_1}, m_2 \in M_{k_2} \quad (12)$$

(R13-14) No solapamiento en la misma máquina (activado solo si ambas asignaciones existen)

$$S_{j'km} \geq C_{jkm} - U(1 - y_{jj'km}) - U(2 - z_{jkm} - z_{j'km}) \quad (13)$$

$$S_{jkm} \geq C_{j'km} - U y_{jj'km} - U(2 - z_{jkm} - z_{j'km}) \quad (14)$$

La  $-U(2 - z_{jkm} - z_{j'km})$  garantiza que las desigualdades se vuelvan no restrictivas cuando alguna de las dos asignaciones  $z_{jkm}$  o  $z_{j'km}$  es 0.

(R15-16) Enlace cantidad – activación

$$\varepsilon z_{jkm} \leq q_{jkm} \leq U z_{jkm} \quad \forall j, k, m \quad (15)$$

y además,

$$z_{jkm} \leq p_{kj} \quad \forall j, k, m \quad (16)$$

Para las etapas de horno ( $k = 20, 27$ ), se añade una cota adicional de capacidad:

$$q_{jkm} \leq 1000 z_{jkm} \quad \forall j \in J, k \in \{20, 27\}, m \in M_k \quad (17)$$

(R18) Orden activación

$$y_{jj',km} \leq z_{jkm}, \quad y_{jj',km} \leq z_{j'km} \quad \forall j \neq j', k, m \quad (18)$$

(R19) Makespan

$$C_{\max} \geq C_{jkm} \quad \forall j, k, m \quad (19)$$

Nota: Las etapas  $k = 20$  y  $k = 27$  corresponden a los hornos de alta y baja temperatura. La etapa de horno tiene un tiempo de procesado fijo, pero cuando la cantidad de producto supera la capacidad del horno, hacen falta varios ciclos. Por eso se usa la variable binaria  $z_{jkm}$  para activar cada ciclo y se añade una restricción de capacidad para limitar las unidades que puede haber a la vez.

Cuando ya se tuvo definido el modelo matemático, el siguiente paso fue programar el código en Python. Con dos tipos de productos se fueron añadiendo etapas al ver que el tiempo de ejecución era bajo, hasta que se implementaron todas las etapas de la planta de

producción. A partir de aquí, el siguiente paso fue añadir productos con distintas rutas de proceso.

Se realizaron dos tipos de pruebas. La primera consistía en incrementar la demanda en 100 pares de producto por cada nuevo producto que se incorporaba. Por tanto, se empezó con 200 y cada producto nuevo sumaba 100 pares más. Se puede observar que el tiempo de ejecución aumenta de manera exponencial con el número de productos, hasta que con 10 productos ya no se ha conseguido una solución óptima en un tiempo razonable, como se puede observar en la Tabla 2.

**Tabla 2**

*Resultados del segundo modelo matemático para demanda creciente*

N.º de productos	N.º pares totales	Tiempo ejecución (s)	Makespan (min)	¿Solución óptima?
2	200	0,56	1552,40	Sí
3	300	0,90	1552,40	Sí
4	400	3,52	1552,40	Sí
5	500	2,95	1552,40	Sí
6	600	4,99	1552,40	Sí
7	700	16,63	1552,40	Sí
8	800	71,76	1571,80	Sí
9	900	672,94	1696,80	Sí
10	1000	+84500		No

Con respecto al primer modelo, este supone un avance claro. El anterior solo permitía trabajar con dos productos y dos etapas, y una demanda muy baja. En cambio, ahora es posible incluir todas las etapas e ir añadiendo productos con mayores demandas.

Por otro lado, se realizaron pruebas con una demanda fija, independientemente de que se fueran añadiendo productos nuevos. Esto provocaba que, aunque se tuviera la misma demanda, al cambiar las rutas de los productos el tiempo de ejecución aumentase, ya que el programa efectuaba más cálculos. Aun así, fue viable resolver hasta 8 productos en un tiempo razonable de cálculo, como se puede comprobar en la Tabla 3.

**Tabla 3***Resultados del segundo modelo matemático para demanda fija*

N.º de productos	N.º pares totales	Tiempo ejecución (s)	Makespan (min)	¿Solución óptima?
2	2000	0,48	15502,40	Sí
3	2000	1,54	10357,90	Sí
4	2000	2,10	7751,80	Sí
5	2000	3,10	6202,40	Sí
6	2000	7,04	5190,70	Sí
7	2000	13,29	4428,30	Sí
8	2000	88,88	3944,40	Sí
9	2000	+ 67280		No

En conclusión, tanto con demanda creciente o demanda fija, el modelo llega a un punto en el que se vuelve muy costoso computacionalmente. Pese a ello, el resultado que se consigue con este segundo modelo es más realista para reproducir una planta de producción de calzado que el primer modelo.

Igualmente se consideró que el tiempo de ejecución podía ser menor, así que se optó por utilizar el mismo modelo matemático pero con una sola máquina por etapa, para reducir la carga computacional. Esto es lo que se desarrollará en el siguiente apartado.

#### 4.4 SIMPLIFICACIÓN DEL SEGUNDO MODELO MATEMÁTICO: POR BLOQUES CON UNA MÁQUINA POR ETAPA

El modelo anterior ofrecía resultados buenos, aunque se podía mejorar el tiempo de ejecución simplificando la estructura. Es por eso por lo que en este apartado se decide modificar el modelo para que solo haya una máquina por etapa, es decir, que cada etapa se trate como un único recurso en lugar de varios en paralelo. De este modo, los tiempos de proceso de cada tipo de producto se ajustan dividiéndose por el número de máquinas que tenía esa etapa. Así se mantiene la lógica de la planificación, resolviéndose en menos tiempo.

En esta simplificación se usa la misma formulación matemática que en el segundo modelo, y el objetivo sigue siendo el mismo: minimizar  $C_{\max}$ . No cambian las

restricciones, pero sí la forma de representarlo ya que ahora solo hay una máquina por etapa. Por eso, el parámetro  $r_{jkm}$  de tiempo de proceso se divide entre el número de máquinas que había antes en cada etapa. Las etapas de los hornos se mantienen igual, debido a que su funcionamiento se basa en la capacidad total que pueden procesar (en pares de zapatos), y no en el número de máquinas.

Como los conjuntos, parámetros, variables y restricciones son exactamente iguales a las del modelo anterior, no se vuelven a incluir en esta sección. Se asumen las del apartado anterior (modelo 2), ya que solo cambia la forma en que se interpreta el número de máquinas por etapa.

De este modo, se avanza hacia los resultados obtenidos. Primeramente, se ejecutó el programa aumentando la demanda en 100 unidades por cada nuevo tipo de producto que se añadía, y en comparación con el modelo previo, se lograron mejoras en los tiempos de ejecución. Mientras que con el modelo 2 los cálculos eran razonables hasta 9 productos, con esta simplificación se alcanza a procesar hasta 14 productos, como se muestra en la Tabla 4.

**Tabla 4**

*Resultados simplificación de una máquina por etapa con demanda creciente*

N.º de productos	N.º pares totales	Tiempo ejecución (s)	Makespan (min)	¿Solución óptima?
2	200	0,04	1550,00	Sí
3	300	0,13	1550,00	Sí
4	400	0,16	1550,00	Sí
5	500	0,15	1550,00	Sí
6	600	0,17	1550,00	Sí
7	700	0,34	1550,00	Sí
8	800	0,37	1570,00	Sí
9	900	1,93	1695,00	Sí
10	1000	4,10	1830,00	Sí
11	1100	7,24	1838,33	Sí
12	1200	79,29	1878,33	Sí
13	1300	286,28	1931,67	Sí
14	1400	3262,67	2028,33	Sí
15	1500	+68905		No

Cabe señalar que en la Tabla 4, por ejemplo, para el caso de 2 a 7 productos se observa un makespan de 1550 minutos, mientras que en la Tabla 2 del segundo modelo matemático el makespan es de 1552,40 minutos. Esta diferencia se debe a que, en el modelo 2, algunas etapas trabajan en paralelo con varias máquinas, tomando el tiempo de finalización como el máximo de todas ellas, mientras que en el modelo simplificado cada etapa se ejecuta en una sola máquina.

En el segundo ensayo, al igual que con el modelo 2, se fijaba una demanda y si incrementaba el número de productos. Cada uno de ellos con una ruta distinta para simular la fábrica de calzado.

**Tabla 5**

*Resultados de la simplificación de una máquina por etapa con demanda fija de 2000 pares*

N.º de productos	N.º pares totales	Tiempo ejecución (s)	Makespan (min)	¿Solución óptima?
2	2000	0,04	15500,00	Sí
3	2000	0,09	10350,05	Sí
4	2000	0,10	7750,00	Sí
5	2000	0,15	6200,00	Sí
6	2000	0,30	5184,95	Sí
7	2000	0,19	4422,75	Sí
8	2000	0,90	3942,50	Sí
9	2000	1,85	3797,50	Sí
10	2000	3,29	3660,00	Sí
11	2000	7,12	3341,25	Sí
12	2000	85,76	3141,20	Sí
13	2000	147,89	2990,27	Sí
14	2000	982,15	2911,43	Sí
15	2000	5310,19	2901,73	Sí

Como se aprecia en la Tabla 5, estos resultados ya mejoran bastante respecto al segundo modelo. Si los comparamos, podemos ver que con la demanda fija de 2000 pares pasamos de 8 productos a 15 productos que encuentran solución óptima en un tiempo lógico.

Ya que con 15 tipos de productos se consigue buen resultado, se decide simular una alta carga productiva, aumentando la demanda fija a 7500 pares. Se hace para analizar cómo

se comporta el modelo frente a volúmenes de producción más altos y estudiar la escalabilidad del programa.

**Tabla 6**

*Resultados de la simplificación de una máquina por etapa con demanda fija de 7500 pares*

N.º de productos	N.º pares totales	Tiempo ejecución (s)	Makespan (min)	¿Solución óptima?
8	7500	0,47	14767,47	Sí
9	7500	2,19	14154,30	Sí
10	7500	4,96	13742,50	Sí
11	7500	7,71	12541,55	Sí
12	7500	14,60	11765,83	Sí
13	7500	188,61	11140,80	Sí
14	7500	1131,27	10884,33	Sí
15	7500	+67500		No

En estas condiciones, al aumentar la demanda y teniendo 15 productos, el programa no encuentra una solución óptima en un tiempo razonable. Puesto que en las fábricas reales esta demanda está más cerca de la realidad, debemos reconsiderar si esta simplificación sirve para un problema práctico. Por eso surge la necesidad de explorar otras estrategias, como usar heurísticas basadas en permutaciones, que se verán en el siguiente apartado para representar mejor el funcionamiento de la planta.

#### 4.5 SIMULACIÓN HEURÍSTICA BASADA EN ÓRDENES Y PERMUTACIONES

Este cambio en la estrategia, en lugar de un modelo matemático, se basa en dos motivos. Primero, los modelos matemáticos (par a par y por bloques) son muy costosos computacionalmente con 15 productos o altas demandas. En segundo lugar, esto surge de la necesidad de encontrar soluciones razonables en un tiempo adecuado. Como toda heurística, no garantiza resultados óptimos, pero ofrece soluciones de buena calidad.

La heurística propuesta se apoya en los dos modelos matemáticos descritos anteriormente. Por un lado, del primer modelo, se utiliza el concepto par a par, y del segundo modelo incorpora el enfoque por bloques. En la práctica, se fuerza el algoritmo a que trabaje par a par, pero obliga a que termine todos los pares del producto que procesa antes de pasar

al siguiente tipo de producto. Por tanto, cada par se puede procesar individualmente en las máquinas y etapas, pero se reduce el número total de variables a calcular ya que se trata como un bloque toda la demanda de ese producto.

El objetivo sigue siendo el mismo: minimizar el tiempo total, o makespan, para reducir el tiempo de producción de la fábrica.

$$\min C_{max}$$

En este caso, en lugar de plantear un modelo matemático completo, se propone una heurística secuencial que consta de tres fases: cálculo del orden inicial, simulación y mejora del orden mediante permutaciones. A continuación se resume su funcionamiento lógico mediante un pseudocódigo conceptual.

#### FASE 1: CÁLCULO DEL ORDEN INICIAL (HEURÍSTICA SIGMA)

**Objetivo:** Determinar en qué orden se introducirán los productos en la planta de producción.

**Pasos:**

**1. Para cada tipo de producto:**

-Identificar todas las etapas por las que debe pasar.

-Para cada etapa que realiza:

- Obtener el tiempo de proceso en esa etapa.
- Obtener el número de máquinas disponibles en esa etapa.
- Calcular la carga de la etapa como:

$$carga_{j,k} = \frac{\text{tiempo de proceso}_{j,k} \times demanda_j}{\text{número de máquinas en la etapa } k} \quad (20)$$

donde  $j$  es el producto,  $k$  la etapa y  $d_j$  la demanda.

-De todas las cargas calculadas para un producto, se toma la máxima como la carga del producto.

## 2. Ordenar todos los productos:

-Los productos se ordenan de menor a mayor carga.

-Resultado: secuencia inicial de productos  $[P_{(1)}, P_{(2)}, P_{(3)}, \dots, P_{(n)}]$ .

Justificación: Los productos con menor carga crítica se priorizan para reducir cuellos de botella.

## FASE 2: SIMULACIÓN DE LA PRODUCCIÓN

**Objetivo:** Evaluar el tiempo total ( $C_{max}$ ) que tarda en completarse toda la producción con un orden dado.

### Pasos:

#### 1. Inicialización:

-Tiempo actual = 0.

-Todas las máquinas están libres.

-Crear cola con todos los productos según el orden establecido.

-Cada producto comienza en su primera etapa.

#### 2. Bucle principal (mientras queden productos por completar):

##### a) Liberar máquinas:

-Revisar todas las etapas.

-Si un producto terminó su tiempo de proceso:

- Si tiene más etapas pendientes: enviarlo a su siguiente etapa.
- Si era su última etapa: marcarlo como completado.

##### b) Asignar productos a máquinas:

-Para cada producto en cola (según el orden):

- Verificar si hay una máquina libre en su etapa actual.
- Si hay máquina libre:
  - Asignar el producto a esa máquina.

- Registrar el tiempo de inicio.
- Calcular tiempo de finalización = tiempo\_actual + duración\_proceso.
- Marcar la máquina como ocupada hasta ese tiempo.
- Si no hay máquina libre:
  - El producto permanece en espera.

**c) Avanzar el tiempo:**

- Encontrar el siguiente evento (siguiente producto que termina).
- Actualizar tiempo\_actual a ese momento.

**3. Calcular resultado:**

- $C_{max}$  = tiempo de finalización del último producto completado.

**FASE 3: MEJORA DEL ORDEN (BÚSQUEDA TABÚ CON PERMUTACIONES)**

**Objetivo:** Mejorar el orden inicial probando diferentes secuencias de productos.

**Pasos:**

**1. Configuración inicial:**

- Orden actual = orden obtenido en Fase 1.
- Mejor orden encontrado = orden actual.
- Mejor  $C_{max}$  = resultado de simular el orden actual.
- Lista de movimientos prohibidos (tabú) = vacía.
- Contador de intentos = 0.

**2. Mientras no se alcance el límite de intentos  $I_{max}$ :**

$I_{max}$  representa el número máximo de iteraciones permitidas sin encontrar mejora, definido numéricamente por el usuario. Al mismo tiempo, la lista tabú tiene un tamaño máximo  $T_{max}$ , también elegido por el usuario.

**a) Generar nuevo orden:**

- Seleccionar dos posiciones aleatorias  $i, j$  en el orden actual.
- Verificar que intercambiar las posiciones  $(i, j)$  no esté en la lista tabú.
- Si está prohibido: intentar con otro par de posiciones.
- Si no está prohibido: crear nuevo\_orden intercambiando elementos en posiciones  $i$  y  $j$ .

**b) Evaluar nuevo orden:**

- Simular la producción con nuevo\_orden (repetir Fase 2).
- Obtener nuevo\_Cmax.

**c) Decidir si aceptar el cambio:**

-Si nuevo\_Cmax < mejor\_Cmax (mejora real):

- Actualizar mejor\_orden = nuevo\_orden.
- Actualizar mejor\_Cmax = nuevo\_Cmax.
- Vaciar lista tabú (permitir explorar nuevamente).
- Usar este nuevo orden como base para siguientes iteraciones.

-Si nuevo\_Cmax  $\geq$  mejor\_Cmax (no mejora):

- Intensificar búsqueda: hacer 3 intentos adicionales.
  - Para cada intento (máximo 3):
    - Generar otra permutación del orden que no mejoró.
    - Simular y evaluar.
    - Si alguna mejora el mejor\_Cmax: actualizar y limpiar lista tabú.
- Si ninguno de los 3 intentos mejora:
  - Añadir el intercambio  $(i, j)$  a la lista tabú.
  - Este intercambio queda prohibido temporalmente.
  - Continuar búsqueda desde el mejor orden conocido.

**d) Incrementar contador de intentos**

**3. Resultado final:**

- Devolver el mejor orden encontrado.
- Devolver el mejor  $C_{max}$  alcanzado.

Además, hay una serie de consideraciones importantes que conviene mencionar. En primer lugar, la lista tabú sirve para evitar probar intercambios que se han explorado recientemente. Se permite de hasta tres intentos adicionales sin mejora para escapar de óptimos locales.

En cuanto al criterio de mejora, este garantiza que se acepte un nuevo orden si  $C_{max}$  es estrictamente menor que el actual. Si la diferencia es muy pequeña, menor que un umbral ( $\epsilon$ ), se considera empate. Esto ayuda a evitar supuestas mejoras provocadas por errores numéricos.

El método heurístico utiliza una estrategia de búsqueda que combina dos enfoques. Por un lado, la diversificación, con la lista tabú, que fuerza a explorar nuevas zonas. Por otro, la intensificación refleja los tres intentos adicionales cercanos al orden actual. Con este equilibrio, se explora lo suficiente sin alejarse demasiado de soluciones prometedoras.

Finalmente, el algoritmo puede finalizar por dos motivos: cuando se alcanza el número máximo de intentos o cuando ya no queden permutaciones válidas que se puedan probar.

El pseudocódigo describe la heurística de forma independiente a la implementación en Python, ya que se centra en los pasos lógicos del método. El código de Python se puede encontrar en el apéndice.

Una vez hecha esta descripción general, es momento de comentar los resultados más relevantes. La heurística obtiene el mismo makespan que el primer modelo matemático para el caso de dos etapas y dos productos. Pero esto no significa que los dos enfoques sean equivalentes. El modelo, teóricamente, debería ser capaz de encontrar soluciones iguales o mejores. El problema es que solo puede ejecutarse con pocos pares, como se mostraba en la Tabla 1, ya que el tiempo de cálculo crecía rápidamente. Por tanto, no es posible comprobar su ventaja, mientras que la heurística sí permite hallar soluciones para problemas mucho más grandes.

En cambio, cuando se compara con el segundo modelo, se observan mejores resultados en el tiempo de producción total. Es tanto así, que el makespan disminuye en torno a un 75% en el caso de una demanda de 200 unidades y 2 productos, como se observa en la Tabla 7 en comparación con la Tabla 2.

**Tabla 7**

*Resultados de la heurística para demanda creciente*

N.º de productos	N.º pares totales	Tiempo ejecución (s)	Makespan (min)
2	200	2,14	346,20
3	300	5,11	370,00
4	400	13,45	454,50
5	500	68,03	548,60
6	600	316,77	648,60
7	700	345,21	748,60
8	800	853,07	848,60
9	900	1324,09	948,60
10	1000	2013,10	1048,60
11	1100	3313,39	1048,60
12	1200	4852,48	1148,60
13	1300	4776,65	1248,60
14	1400	8240,49	1348,60
15	1500	10117,51	1448,60

Esto sucede porque la heurística trabaja par a par, en lugar de por bloques de producto, como hace el modelo matemático 2. Aunque mejora el makespan, los tiempos de ejecución son algo elevados, ya que el método prueba todas las combinaciones posibles y realiza hasta tres permutaciones adicionales por intento antes de finalizar la ejecución del programa.

En consecuencia, se han introducido algunos ajustes para mejorar el rendimiento de la heurística y evitar ejecuciones tan largas. En primer lugar, se ha añadido un criterio de parada basado en el estancamiento, que consiste en contar intentos consecutivos sin mejora. Si llega a veinte, la búsqueda se detiene pero el contador se reinicia cada vez que aparece una mejora, ya sea de forma directa o dentro de las permutaciones adicionales.

Al mismo tiempo, se ha limitado el tamaño de la lista tabú. En la versión inicial, la lista crecía sin control y esto dificultaba generar nuevas permutaciones válidas, además de ralentizar la búsqueda. Por eso, ahora solo se conservan los veinte movimientos más recientes y, cuando se supera ese límite, se eliminan los más antiguos.

Con estos dos cambios se observa una mejoría en el tiempo de ejecución del programa respecto a los valores de la Tabla 7. Aunque en la Tabla 8 los tiempos no disminuyen demasiado, el rendimiento es bueno, sobre todo si tenemos en cuenta que trabaja con un máximo de veinte intentos de mejora y, aun así, se obtienen los mismos resultados.

**Tabla 8**

*Resultados heurística con límite de intentos con demanda creciente*

N.º de productos	N.º pares totales	Tiempo ejecución (s)	Makespan (min)
2	200	3,12	346,20
3	300	6,57	370,00
4	400	25,78	454,50
5	500	71,48	548,60
6	600	182,96	648,60
7	700	308,47	748,60
8	800	686,03	848,60
9	900	846,68	948,60
10	1000	1215,69	1048,60
11	1100	2555,25	1048,60
12	1200	2822,34	1148,60
13	1300	2293,56	1248,60
14	1400	2846,55	1348,60
15	1500	3675,02	1448,60

Se puede hacer una comparación entre el número de intentos y el tiempo de ejecución que ha tardado el programa en llegar al mismo makespan. La Tabla 9 muestra que limitar los intentos cuando no hay mejoras y acotar el tamaño de la lista tabú reduce los intentos y el tiempo de ejecución sin afectar al resultado final.

**Tabla 9***Comparación intentos y tiempo de ejecución*

N.º de productos	Tiempo ejecución caso 1 (s)	N.º de intentos caso 1	Tiempo ejecución caso 2 (s)	N.º de intentos caso 2	Makespan (min)
2	2,14	1	3,12	1	346,20
3	5,11	2	6,57	2	370,00
4	13,45	4	25,78	4	454,50
5	68,03	10	71,48	10	548,60
6	316,77	15	182,96	15	648,60
7	345,21	20	308,47	20	748,60
8	853,07	34	686,03	30	848,60
9	1324,09	38	846,68	29	948,60
10	2013,10	44	1215,69	29	1048,60
11	3313,39	74	2555,25	49	1048,60
12	4852,48	86	2822,34	51	1148,60
13	4776,65	71	2293,56	33	1248,60
14	8240,49	96	2846,55	35	1348,60
15	10117,51	96	3675,02	37	1448,60

Estas mejoras resultan útiles para evitar que la búsqueda se prolongue innecesariamente, mejorando así el rendimiento de la heurística en problemas con muchos productos.

En resumen, con esta heurística es posible simular de manera realista una planta de producción de calzado y, al mismo tiempo, alcanzar la optimización que se perseguía con los modelos matemáticos, completando así el proceso de planificación y secuenciación desarrollado a lo largo de este capítulo.

## 5. CONCLUSIONES

El objetivo de este trabajo era estudiar diferentes formas de optimizar la planificación de una planta de producción de calzado. Se tuvo en cuenta que cada producto seguía su propia ruta, con tiempos diferentes en cada etapa y un número concreto de máquinas disponibles. Por tanto, se probaron varios modelos matemáticos y un método heurístico para obtener soluciones viables.

El modelo inicial permitía planificar cada par de zapatos por separado y daba resultados correctos, pero solo funcionaba con demandas muy pequeñas, porque el tiempo de ejecución aumentaba muy rápidamente.

En un segundo planteamiento se planificaba la producción por bloques de producto, es decir, agrupaba todos los pares de un mismo tipo, lo cual reducía el número de variables. Pero pese a que se pudo añadir todas las etapas de la fábrica ficticia, al aumentar el número de productos también lo hizo el tiempo de cálculo de la solución óptima.

Por ese motivo se realizó una simplificación del segundo modelo matemático, dejando solo una máquina por etapa y ajustando los tiempos de proceso según el número de máquinas original. Esta idea ayudó a reducir la carga computacional, pero no evitó que el problema se volviera demasiado grande cuando se incrementaba la demanda.

Posteriormente, se optó por buscar otro enfoque: una heurística basada en permutaciones que permitiera obtener resultados en un tiempo razonable. Esta heurística combinaba las ideas anteriores, ya que trabajaba par a par pero sin mezclar productos entre sí. Aunque no garantiza la solución óptima, los resultados fueron mejores que con los modelos matemáticos. El makespan disminuía y los tiempos de ejecución se consideraban razonables.

En conjunto, este trabajo ha permitido ver hasta dónde pueden llegar los modelos matemáticos exactos y en qué punto dejan de ser útiles por su carga computacional.

También ha mostrado que, en un entorno real, una heurística bien diseñada puede resultar más práctica, ya que ofrece soluciones rápidas y manejables.

De cara a futuros trabajos, se pueden explorar nuevas reglas iniciales en la heurística, profundizar en técnicas de mejora más amplias o añadir restricciones operativas, como los cambios entre modelos de calzado o la gestión de turnos. También sería interesante estudiar cómo cambia el makespan si se modifican máquinas o capacidades, y crear una herramienta práctica que permita utilizar la heurística directamente en planta.



## BIBLIOGRAFÍA

Cabral, L. (2022). *Introducción a la organización industrial* (2.<sup>a</sup> ed.). Antoni Bosch Editor. <https://elibro.net/es/ereader/bibliotecaumh/215916?page=19>

Casiello, F. A. (2021). *Optimización de operaciones: De producción y de servicio*. Editorial Académica Española.

Chankov, S., Hütt, M. T., & Bendul, J. (2016). Synchronization in manufacturing systems: Quantification and relation to logistics performance. *International Journal of Production Research*, 54(20), 6033–6051.

Gurobi Optimization, LLC. (2024). *Gurobi Optimizer* (Version 12.0.2) [Computer software]. <https://www.gurobi.com>

Hazewinkel, M. (Ed.). (2021). *Makespan*. En *Encyclopaedia of Mathematics*. Springer. Recuperado el 7 de noviembre de 2025, de <https://encyclopediaofmath.org/wiki/Makespan>

Heizer, J., Render, B., & Parra, J. L. M. (2007). *Dirección de la producción y de operaciones: decisiones estratégicas*. Madrid: Pearson educación.

Los Santos, I. S. (2010). *Logística y operaciones en la empresa*. ESIC Editorial.

MacCarthy, B. (2006). Organizational, systems and human issues in production planning, scheduling and control. In J. W. Herrmann (Ed.), *Handbook of production scheduling* (pp. 59–90). Springer US.

Meilanitasari, P., & Shin, S. J. (2021). A review of prediction and optimization for sequence-driven scheduling in job shop flexible manufacturing systems. *Processes*, 9(8), 1391.

Mora, C. M., & de Lucas, F. M. (2017). La estrategia de retorno de la industria española: El caso del sector calzado en Alicante, su importancia y determinantes. *Studies of Applied Economics*, 35(3), 777–800.

Muliono, R., Silviana, N. A., & Novita, N. (2024). Involvement of Various Selection Methods for Genetic Algorithms in Determining the Optimal Production Schedule Problem. *JOIV: International Journal on Informatics Visualization*, 8(4), 2494-2501.

Oksuz, M. K., Buyukozkan, K., & Satoglu, S. I. (2017). U-shaped assembly line worker assignment and balancing problem: A mathematical model and two meta-heuristics. *Computers & Industrial Engineering*, *112*, 246-263.

Python Software Foundation. (2024). *Python* (Version 3.13.5) [Computer software]. <https://www.python.org>

Rezig, S., Ezzeddine, W., Turki, S., & Rezg, N. (2020). Mathematical model for production plan optimization—A case study of discrete event systems. *Mathematics*, *8*(6), 955.

Saraçoğlu, İ., Süer, G. A., & Gannon, P. (2021). Minimizing makespan and flowtime in a parallel multi-stage cellular manufacturing company. *Robotics and Computer-Integrated Manufacturing*, *72*, 102182.

Süer, G. A., Subramanian, A., & Huang, J. (2009). Heuristic procedures and mathematical models for cell loading and scheduling in a shoe manufacturing company. *Computers & Industrial Engineering*, *56*(2), 462-475.



## APÉNDICE. CÓDIGO PYTHON DE LA HEURÍSTICA

```
import time
from collections import deque, defaultdict
import math
import random

# Medir tiempo de ejecución del programa
inicio = time.time()
random.seed(0) # Para que el orden de permutaciones sea el mismo cada
vez que ejecute

# ===== CLASES PRINCIPALES =====

class Maquina: # Representa una máquina individual. Lleva control de
cuándo estará disponible para procesar el siguiente producto

    def __init__(self):
        self.disponible_en = 0.0 # Momento en que la máquina queda libre

    def esta_libre(self, t_actual): # Verifica si la máquina está
disponible en el tiempo actual
        return t_actual >= self.disponible_en

    def asignar(self, t_actual, duracion): # Asigna un producto a la
máquina. Devuelve el tiempo de inicio y fin del procesamiento
        inicio = max(t_actual, self.disponible_en)
        fin = inicio + duracion
        self.disponible_en = fin # Actualiza disponibilidad
        return inicio, fin

class Etapa: # Representa una etapa de producción con una o más máquinas
paralelas

    def __init__(self, id, n_maquinas=1, tiempo_fijo=None):
        self.id = id
        self.maquinas = [Maquina() for _ in range(n_maquinas)]
        self.en_proceso = [] # Lista de productos siendo procesados
        self.tiempo_fijo = tiempo_fijo # Para etapas de hornos

    def asignar_producto(self, producto, t_actual, duracion): # Intenta
asignar un producto a alguna máquina disponible en esta etapa

        for maquina in self.maquinas: # Devuelve (True, tiempo_fin) si se
asigna, o (False, None) si no hay máquinas libres
            if maquina.esta_libre(t_actual):
```

```

        inicio, fin = maquina.asignar(t_actual, duracion)
        self.en_proceso.append((producto, fin))
        return True, fin
    return False, None

# ===== FUNCIÓN DE ORDENAMIENTO =====

def calcular_orden(rutas, tiempos, demanda, etapas_info): # Calcula el
orden inicial de productos usando la heurística 'sigma'

    productos = list(demanda.keys())
    etapas = list(etapas_info.keys())
    n_maquinas = {et: etapas_info[et]["n_maquinas"] for et in etapas}

    cargas = {}
    for j in productos:
        cargas_j = []
        # Calcula la carga en cada etapa de la ruta del producto
        for k in rutas[j]:
            if k in tiempos[j]:
                carga_k = tiempos[j][k] * (demanda[j] / n_maquinas[k])
            else:
                carga_k = 0
            cargas_j.append(carga_k)

        # La carga del producto es el máximo entre todas sus etapas
        cargas[j] = max(cargas_j) if cargas_j else 0

    # Ordena productos por carga ascendente
    orden = sorted(productos, key=lambda j: cargas[j])
    return orden, cargas

# ===== SIMULADOR DE PRODUCCIÓN =====

class Simulador: # Simula el proceso de producción completo para un orden
dado de productos

    def __init__(self, rutas, tiempos, demanda, etapas_info, orden=None):
        self.rutas = rutas # Ruta de etapas por cada tipo de producto
        self.tiempos = tiempos # Tiempos de procesamiento
        self.demanda = demanda # Cantidad a producir de cada tipo
        self.etapas_info = etapas_info # Configuración de etapas
        self.etapas = {et: Etapa(et, **info) for et, info in
etapas_info.items()}
        self.tiempo = 0.0 # Reloj de simulación
        self.enCola = deque() # Cola de productos esperando
        self.en_proceso = defaultdict(list)

```

```

self.completados = {} # Registro de productos terminados
self.productos_restantes = sum(demanda.values())
self.orden = orden # Orden de secuenciación a usar

def cargar_productos(self): # Carga todos los productos a la cola
inicial según el orden especificado. Si no hay orden, usa sigma

    if self.orden is None:
        orden, cargas = calcular_orden(self.rutas, self.tiempos,
self.demanda, self.etapas_info)
    else:
        orden = self.orden
        cargas = None

    # Añade todos los productos a la cola
    for tipo in orden:
        for i in range(self.demanda[tipo]):
            self.enCola.append((tipo, i), 0) # (producto,
posición_en_ruta)
    return orden, cargas

def avanzar_etapas(self): # Revisa todas las etapas y mueve productos
que han terminado a la siguiente etapa o a completados

    for etapa in self.etapas.values():
        nuevos = []
        # Identifica productos que terminaron en esta etapa
        for producto, fin in etapa.en_proceso:
            if fin <= self.tiempo:
                nuevos.append(producto)

        # Actualiza lista de productos en proceso
        etapa.en_proceso = [(p, f) for (p, f) in etapa.en_proceso if
f > self.tiempo]

    # Procesa productos terminados
    for producto in nuevos:
        tipo, _ = producto
        ruta = self.rutas[tipo]
        idx = self.posiciones[producto] + 1

        if idx < len(ruta):
            # Pasa a la siguiente etapa
            self.enCola.appendleft((producto, idx))
        else:
            # Producto completado
            self.completados[producto] = self.tiempo
            self.productos_restantes -= 1

```

```

def correr(self, mostrar=True): # Ejecuta la simulación completa

    self.posiciones = {}
    orden, cargas = self.cargar_productos()

    # Registro del primer instante que cada tipo de producto entra a
    # cada etapa
    primeros_en_etapa = defaultdict(dict)

    # Bucle principal de simulación
    while self.productos_restantes > 0:
        self.avanzar_etapas()

        # Intenta asignar productos de la cola a etapas disponibles
        nuevaCola = deque()
        while self.enCola:
            producto, pos = self.enCola.popleft()
            tipo = producto[0]
            etapa = self.rutas[tipo][pos]
            self.posiciones[producto] = pos

            # Determina duración del procesamiento
            if self.etapas[etapa].tiempo_fijo is not None:
                duracion = self.etapas[etapa].tiempo_fijo
            else:
                duracion = self.tiempos[tipo][etapa]

            # Intenta asignar a la etapa
            asignado, fin =
self.etapas[etapa].asignar_producto(producto, self.tiempo, duracion)

            # Registra primer instante del tipo en esta etapa
            if asignado and tipo not in primeros_en_etapa[etapa]:
                primeros_en_etapa[etapa][tipo] = round(self.tiempo,
2)

            # Si no se pudo asignar, vuelve a la cola
            if not asignado:
                nuevaCola.append((producto, pos))

        self.enCola = nuevaCola

        # Avanza el tiempo al siguiente evento (siguiente fin de
        # proceso)
        if self.productos_restantes > 0:
            siguientes_fines = [f for etapa in self.etapas.values()
for (_, f) in etapa.en_proceso]
            if siguientes_fines:
                self.tiempo = min(siguientes_fines)

```

```

        else:
            break

    # Calcula el makespan (tiempo de finalización del último
producto)
    cmax = max(self.completados.values())

    # Muestra resultados si se solicita
    if mostrar:
        print("\nPrimer instante por producto en cada etapa:")
        for etapa, productos in primeros_en_etapa.items():
            print(f" Etapa {etapa}:")
            for prod, t in productos.items():
                print(f" Producto {prod} entra en t={t:.1f} min")

    return cmax, primeros_en_etapa, orden

# ===== ALGORITMO DE BÚSQUEDA =====

def generar_permutacion(orden, lista_tabu): # Genera nueva permutación
intercambiando dos productos aleatorios, evitando los de la lista tabú

    n = len(orden)
    intento = 0
    while intento < 10:
        # Selecciona dos posiciones aleatorias
        i, j = random.sample(range(n), 2)
        par = tuple(sorted((orden[i], orden[j])))

        # Verifica que no esté en lista tabú
        if par not in lista_tabu:
            nuevo = orden.copy()
            nuevo[i], nuevo[j] = nuevo[j], nuevo[i] # Intercambia
            return nuevo, par
        intento += 1
    return orden, None

def busqueda_permutaciones(rutas, tiempos, demanda, etapas_info,
max_intentos=100, epsilon=1e-6,
                        max_sin_mejora=20, max_tabu=20): # Algoritmo de
búsqueda local con lista tabú para optimizar la secuenciación

    # Orden inicial usando heurística sigma
    orden_inicial, cargas = calcular_orden(rutas, tiempos, demanda,
etapas_info)
    mejor_orden = orden_inicial

```

```

# Control de órdenes ya explorados
ordenes_visitados = set()
ordenes_visitados.add(tuple(mejor_orden))

# Evalúa orden inicial
sim = Simulador(rutas, tiempos, demanda, etapas_info,
orden=mejor_orden)
mejor_cmax, _, _ = sim.correr(mostrar=False)

# Variables de control del algoritmo
lista_tabu = [] # Intercambios prohibidos temporalmente
intentos = 0
intentos_sin_mejora = 0

print("\n=== INICIO DE BÚSQUEDA DE PERMUTACIONES ===")
print("Cargas calculadas:", {k: round(v, 2) for k, v in
cargas.items()})
print("Orden inicial sigma:", orden_inicial)
print(f"Cmax inicial: {mejor_cmax:.10f}")
print(f"Umbral de mejora (epsilon): {epsilon}")
print(f"Criterio de parada: {max_sin_mejora} intentos sin mejora")
print(f"Tamaño máximo lista tabú: {max_tabu}\n")

# Bucle principal de búsqueda
while intentos < max_intentos and intentos_sin_mejora <
max_sin_mejora:
    # Genera nueva permutación
    nuevo_orden, par = generar_permutacion(mejor_orden,
set(lista_tabu))
    if par is None:
        print("No hay más permutaciones válidas disponibles.")
        break

    # Evita repetir órdenes ya probados
    if tuple(nuevo_orden) in ordenes_visitados:
        print(f" [Saltando] Orden ya visitado previamente")
        continue
    ordenes_visitados.add(tuple(nuevo_orden))

    # Evalúa la nueva permutación
    sim = Simulador(rutas, tiempos, demanda, etapas_info,
orden=nuevo_orden)
    cmax, _, _ = sim.correr(mostrar=False)

    diferencia = cmax - mejor_cmax
    print(f"\nIntento {intentos + 1}: prueba orden {nuevo_orden}")
    print(f" Cmax = {cmax:.10f}")
    print(f" Intentos sin mejora:
{intentos_sin_mejora}/{max_sin_mejora}")

```

```

# Verifica si hay mejora significativa
if mejor_cmax - cmax > epsilon:
    # Mejora encontrada
    mejora = mejor_cmax - cmax
    print(f"  ✓ ¡MEJORA! Anterior: {mejor_cmax:.10f}")
    print(f"                Nuevo:   {cmax:.10f}")
    print(f"                Δ = {mejora:.10f}")
    print(f"  Se limpia lista tabú y se reinicia contador de
estancamiento.\n")
    mejor_cmax = cmax
    mejor_orden = nuevo_orden
    lista_tabu.clear()
    intentos_sin_mejora = 0
else:
    # No mejora: incrementa contador de estancamiento
    intentos_sin_mejora += 1

    if abs(diferencia) < epsilon:
        print(f"  ≈ Igual (diferencia despreciable:
{diferencia:.2e})")
    else:
        print(f"  X No mejora (peor por: {diferencia:.10f}")

# Intensificación: prueba 3 permutaciones adicionales desde
esta
print("  Probando hasta 3 permutaciones adicionales...")
mejora_local = False
orden_temp = nuevo_orden
intentos_locales_validos = 0

for intento_local in range(3):
    orden_temp_candidato, par_temp =
generar_permutacion(orden_temp, set(lista_tabu))
    if par_temp is None:
        print(f"    [{intento_local + 1}] No hay más
permutaciones disponibles")
        break

    # Evita repeticiones
    if tuple(orden_temp_candidato) in ordenes_visitados:
        continue

    ordenes_visitados.add(tuple(orden_temp_candidato))
    orden_temp = orden_temp_candidato
    intentos_locales_validos += 1

# Evalúa permutación local

```

```

sim_local = Simulador(rutas, tiempos, demanda,
etapas_info, orden=orden_temp)
cmax_local, _, _ = sim_local.correr(mostrar=False)

diferencia_local = cmax_local - mejor_cmax
print(f"      [{intentos_locales_validos}] {orden_temp}")
print(f"      Cmax = {cmax_local:.10f} ( $\Delta$  = {-
diferencia_local:.10f}")

# Verifica mejora local
if mejor_cmax - cmax_local > epsilon:
    mejora_real = mejor_cmax - cmax_local
    print(f"      ✓ ¡MEJORA LOCAL REAL! (mejora de
{mejora_real:.10f}")
    print(f"      Se reinicia contador de
estancamiento.\n")
    mejor_cmax = cmax_local
    mejor_orden = orden_temp
    lista_tabu.clear()
    mejora_local = True
    intentos_sin_mejora = 0
    break

if not mejora_local:
    # Añade a lista tabú (memoria de corto plazo)
    lista_tabu.append(par)
    if len(lista_tabu) > max_tabu:
        par_eliminado = lista_tabu.pop(0) # FIFO
        print(f"      → Se añade {par} a lista tabú y se elimina
{par_eliminado} (más antiguo)")
    else:
        print(f"      → Se añade {par} a lista tabú (tamaño:
{len(lista_tabu)})")
    print()

    intentos += 1

# Reporte final
print("\n=== MOTIVO DE FINALIZACIÓN ===")
if intentos_sin_mejora >= max_sin_mejora:
    print(f" Parada por estancamiento: {intentos_sin_mejora} intentos
consecutivos sin mejora")
elif intentos >= max_intentos:
    print(f" Alcanzado límite máximo de intentos: {max_intentos}")
else:
    print(" No hay más permutaciones válidas disponibles")

print("\n=== RESULTADO FINAL ===")
print(f"Total de intentos realizados: {intentos}")

```

```

print(f"Intentos sin mejora al final: {intentos_sin_mejora}")
print(f"Tamaño final de lista tabú: {len(lista_tabu)}")
print("Mejor orden final:", mejor_orden)
print(f"Mejor Cmax final: {mejor_cmax:.10f}\n")

# Simulación final con detalles
sim_final = Simulador(rutas, tiempos, demanda, etapas_info,
orden=mejor_orden)
sim_final.correr(mostrar=True)

return mejor_orden, mejor_cmax

# ===== DATOS DEL PROBLEMA =====

# Rutas de producción: secuencia de etapas por cada tipo de producto
rutas = {
    1: [1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 31, 32, 33, 34, 35],
    2: [1, 2, 4, 5, 6, 15, 7, 8, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26,
27, 28, 29, 30, 32, 33, 35],
    3: [1, 2, 4, 5, 6, 8, 9, 10, 16, 22, 24, 25, 26, 27, 28, 30, 32, 33,
35],
    4: [1, 2, 4, 5, 8, 9, 10, 16, 22, 24, 25, 26, 27, 28, 32, 33, 35],
    5: [1, 2, 17, 16, 22, 24, 25, 26, 27, 28, 32, 33, 35],
    6: [1, 2, 17, 16, 21, 22, 24, 25, 26, 27, 28, 32, 33, 35],
    7: [1, 8, 9, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 35],
    8: [1, 2, 4, 5, 6, 8, 9, 10, 16, 22, 24, 25, 26, 27, 28, 29, 30, 31,
32, 33, 35],
    9: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 31, 32, 33, 35],
    10:[1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 31, 32, 33, 35],
    11:[1, 2, 5, 7, 9, 8, 15, 10, 14, 21, 25, 26, 27, 28, 31, 32, 33, 34,
35],
    12:[1, 2, 5, 7, 8, 9, 10, 11, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25,
26, 27, 28, 31, 32, 33, 34, 35],
    13:[1, 2, 3, 4, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 31, 32, 33, 34, 35],
    14:[1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 31, 32, 33, 34, 35],
    15:[1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 31, 32, 33, 34, 35],
}

# Tiempos de proceso (producto, etapa)
tiempos = {
    1: {1: 2.0, 2: 1.2, 3: 0.8, 4: 1.0, 7: 1.1, 8: 1.0, 9: 1.0, 10: 1.0,
    11: 1.0, 12: 0.6, 13: 1.2, 14: 0.9, 15: 1.1, 16: 1.0, 17: 0.6,

```

18: 0.4, 19: 0.4, 20: 15, 21: 1.0, 22: 1.0, 23: 1.0, 24: 0.8, 25:  
 0.8,  
 26: 0.6, 27: 20, 28: 0.5, 31: 0.5, 32: 0.9, 33: 0.5, 34: 2.0, 35:  
 1.0},  
 2: {1: 1.2, 2: 0.8, 4: 1.5, 5: 1.0, 6: 0.3, 15: 2.0, 7: 1.8, 8: 1.5,  
 16: 1.2, 18: 0.5, 19: 0.5, 20: 15, 21: 1.0, 22: 1.2, 23: 1.2, 24:  
 1.0,  
 25: 1.0, 26: 0.6, 27: 20, 28: 0.5, 29: 1.0, 30: 0.6, 32: 1.0, 33:  
 0.6,  
 35: 1.0},  
 3: {1: 1.2, 2: 1.0, 4: 1.0, 5: 1.0, 6: 0.3, 8: 1.2, 9: 1.2, 10: 1.3,  
 16: 1.2, 22: 1.0, 24: 0.8, 25: 0.8, 26: 0.6, 27: 20, 28: 0.5, 30:  
 0.7,  
 32: 1.2, 33: 0.5, 35: 1.0},  
 4: {1: 1.0, 2: 0.9, 4: 0.9, 5: 0.8, 8: 1.0, 9: 1.0, 10: 1.1,  
 16: 1.0, 22: 1.0, 24: 0.7, 25: 0.8, 26: 0.6, 27: 20, 28: 0.5, 32:  
 1.0,  
 33: 0.5, 35: 1.0},  
 5: {1:0.9, 2:0.8, 17:0.6, 16:1.0, 22:1.0, 24:0.8, 25:1.0, 26:0.6,  
 27:20,  
 28:0.5, 32:0.9, 33:0.5, 35:1.0},  
 6: {1: 0.8, 2: 0.6, 17: 0.6, 16: 1.0, 21: 1.0, 22: 1.0, 24: 0.8, 25:  
 0.8,  
 26: 0.6, 27: 20.0, 28: 0.5, 32: 0.9, 33: 0.5, 35: 1.0},  
 7: {1:1.0, 8:1.0, 9:1.0, 16:1.0, 18:0.4, 20:15.0, 21:1.0, 22:1.0,  
 24:0.8, 25:0.8,  
 26:0.6, 27:20.0, 28:0.5, 30:0.6, 32:0.9, 33:0.5, 35:1.0},  
 8: {1:1.2, 2:1.0, 4:1.0, 5:1.0, 6:0.3, 8:1.0, 9:1.0, 10:1.2, 16:1.2,  
 22:1.0,  
 24:0.8, 25:0.8, 26:0.6, 27:20.0, 28:0.5, 29:0.7, 30:0.6, 31:0.6,  
 32:1.2, 33:0.5, 35:1.0},  
 9: {1:2.0, 2:1.2, 3:1.0, 4:1.0, 5:1.0, 6:0.3, 7:0.5, 8:1.0, 9:1.0,  
 10:1.2,  
 11:0.8, 12:0.8, 13:1.0, 15:1.0, 16:1.0, 17:0.6, 18:0.4, 19:0.6,  
 20:15.0, 21:1.0,  
 22:1.0, 23:1.0, 24:0.8, 25:0.8, 26:0.6, 27:20.0, 28:0.5, 31:0.6,  
 32:0.9, 33:0.5, 35:1.0},  
 10: {1:2.2, 2:1.3, 4:1.2, 5:1.2, 6:0.4, 7:0.6, 8:1.2, 9:1.2, 10:1.4,  
 11:1.0,  
 13:1.2, 14:1.0, 15:1.0, 16:1.2, 17:0.7, 18:0.5, 19:0.6, 20:15.0,  
 21:1.0,  
 22:1.0, 23:1.0, 24:0.9, 25:0.9, 26:0.7, 27:20.0, 28:0.6, 31:0.6,  
 32:1.2, 33:0.5, 35:1.0},  
 11: {1:1.0, 2:0.6, 5:1.0, 7:0.6, 8:1.0, 9:1.0, 10:1.2, 14:0.8,  
 15:0.5, 21:1.0,  
 25:0.8, 26:0.6, 27:20.0, 28:0.5, 31:0.6, 32:0.9, 33:0.5, 34:0.4,  
 35:1.0},  
 12: {1:1.6, 2:1.0, 5:1.0, 7:0.6, 8:1.0, 9:1.0, 10:1.2, 11:1.0,  
 15:0.5, 16:1.0,

```

    18:0.4, 19:0.6, 20:15.0, 21:1.0, 22:1.0, 23:0.5, 24:0.8, 25:0.8,
26:0.6,
    27:20.0, 28:0.5, 31:0.6, 32:0.9, 33:0.5, 34:0.4, 35:1.0},
    13: {1:2.0, 2:1.2, 3:0.8, 4:1.0, 7:0.5, 8:1.0, 9:1.0, 10:1.2, 12:0.6,
13:1.2,
    14:0.9, 15:1.1, 16:1.0, 17:0.6, 18:0.4, 19:0.6, 20:15.0, 21:1.0,
22:1.0,
    23:1.0, 24:0.8, 25:0.8, 26:0.6, 27:20.0, 28:0.5, 31:0.6, 32:0.9,
33:0.5, 34:2.0, 35:1.0},
    14: {1:2.0, 2:1.2, 3:0.8, 4:1.0, 7:1.1, 8:1.0, 9:1.1, 10:1.3, 11:1.2,
12:0.6,
    13:1.5, 14:0.9, 15:1.1, 16:1.0, 17:0.8, 18:0.4, 19:0.4, 20:15.0,
21:1.0,
    22:1.0, 23:1.0, 24:0.8, 25:0.8, 26:0.6, 27:20.0, 28:0.5, 31:0.6,
32:0.9, 33:0.5, 34:2.0, 35:1.0},
    15: {1:2.4, 2:1.4, 3:0.9, 4:1.2, 7:1.3, 8:1.2, 9:1.3, 10:1.3, 11:1.1,
12:0.7,
    13:1.4, 14:1.0, 15:1.2, 16:1.2, 17:0.7, 18:0.4, 19:0.5, 20:15.0,
21:1.0,
    22:1.0, 23:1.1, 24:0.9, 25:0.8, 26:0.6, 27:20.0, 28:0.5, 31:0.7,
32:1.1, 33:0.5, 34:2.0, 35:1.0},
}

# Demanda por tipo de producto
demanda = {1: 100, 2: 100, 3: 100, 4: 100, 5: 100, 6: 100, 7: 100, 8:
100, 9: 100, 10: 100, 11: 100, 12:100, 13:100, 14: 100, 15: 100}

# Número de máquinas por etapa
etapas_info = {
    1: {"n_maquinas": 6}, 2: {"n_maquinas": 3}, 3: {"n_maquinas": 3}, 4:
{"n_maquinas": 2},
    5: {"n_maquinas": 1}, 6: {"n_maquinas": 1}, 7: {"n_maquinas": 1}, 8:
{"n_maquinas": 2},
    9: {"n_maquinas": 2}, 10: {"n_maquinas": 2}, 11: {"n_maquinas": 2},
12: {"n_maquinas": 1},
    13: {"n_maquinas": 2}, 14: {"n_maquinas": 2}, 15: {"n_maquinas": 2},
16: {"n_maquinas": 3},
    17: {"n_maquinas": 1}, 18: {"n_maquinas": 2}, 19: {"n_maquinas": 2},
20: {"n_maquinas": 100, "tiempo_fijo": 15},
    21: {"n_maquinas": 1}, 22: {"n_maquinas": 1}, 23: {"n_maquinas": 1},
24: {"n_maquinas": 2},
    25: {"n_maquinas": 2}, 26: {"n_maquinas": 2}, 27: {"n_maquinas": 100,
"tiempo_fijo": 20},
    28: {"n_maquinas": 2}, 29: {"n_maquinas": 1}, 30: {"n_maquinas": 1},
31: {"n_maquinas": 2},
    32: {"n_maquinas": 3}, 33: {"n_maquinas": 2}, 34: {"n_maquinas": 2},
35: {"n_maquinas": 2},
}

```

```
# Ejecutar con parámetros optimizados
mejor_orden, mejor_cmax = busqueda_permutaciones(
    rutas, tiempos, demanda, etapas_info,
    max_intentos=100,      # límite absoluto de iteraciones
    epsilon=1e-6,        # mejora mínima para considerar mejora real
    (por defecto: 0.000001)
    max_sin_mejora=20,    # parada por estancamiento (por defecto: 20)
    max_tabu=20          # tamaño máximo de lista tabú (por defecto:
20)
)

fin = time.time() # Se para el contador del tiempo de ejecución
print(f"\nTiempo de ejecución real: {fin - inicio:.2f} segundos")
```

