



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA

Modelo de optimización para la planificación de la producción industrial de raciones para animales

Informe de Proyecto de Grado presentado por

Federico Vallcorba

en cumplimiento parcial de los requerimientos para la graduación de la carrera
de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de
la República

Supervisores

Dr. Ing. Cristina Mayr
MSc. Ing. Agustín López de Lacalle

Montevideo, 22 de julio de 2024



Modelo de optimización para la planificación de la producción industrial de raciones para animales por Federico Vallcorba tiene licencia [CC Atribución 4.0](https://creativecommons.org/licenses/by/4.0/).

INTEGRANTES DEL TRIBUNAL DE DEFENSA DE TESIS DE GRADO

Dr. Gonzalo Perera

Dr. Ing. Claudio Risso

Dr. Ing. Rafael Sotelo

Agradecimientos

Me gustaría agradecer a todos los que contribuyeron en la generación de este documento. En particular, a mis tutores, Agustín y Cristina, por su tiempo y guía a lo largo de todo el proyecto. También agradecer a familiares y amigos por su apoyo brindado a lo largo de la carrera.

Pro último, agradecerle a todos los profesores de la Facultad de Ingeniería por su dedicación y enseñanzas, y a todos los compañeros con los cuales compartí clases y tareas.

Resumen

La ganadería cumple un rol clave en la actividad agropecuaria de Uruguay. Uno de sus principales insumos son las raciones para animales, por lo que su producción termina impactando significativamente en gran parte de los procesos productivos de la actividad agropecuaria del país.

Los alimentos balanceados para animales se producen en distintos formatos y presentaciones. Cada combinación anterior implica la participación de distintos procesos en la planta de elaboración de alimentos. Dado el gran impacto que podría tener la optimización de estos procesos, resulta interesante el estudio de modelos que permitan optimizar el uso de los recursos productivos disponibles.

En el presente trabajo se aborda el problema de asignación de jornales en la planta de producción de alimentos. El problema en concreto es, dada una demanda semanal de distintos tipos de alimentos con distintos tiempos de retiros, minimizar la cantidad de jornales requeridos para satisfacer la demanda en tiempo y forma. En este marco, se creó un modelo de programación lineal que permite la resolución del problema descrito. El modelo representa una planta de elaboración de alimentos balanceados, con sus distintas etapas y las restricciones propias del funcionamiento de la planta.

El modelo se implementó en el lenguaje *Python*, haciendo uso del módulo *DOCPLEX* de IBM, que brinda una interfaz para utilizar las funcionalidades de CPLEX. Una vez implementado, se realizaron múltiples pruebas para verificar el correcto funcionamiento del modelo, además de pruebas para analizar el impacto de los parámetros en el mismo, basándose en datos reales de la industria.

El modelo implementado no busca ser una versión definitiva. En ese sentido, se incluyen diversas líneas de trabajo futuro orientadas a continuar y expandir el trabajo presentado.

Palabras clave: Alimentos balanceados, Raciones, optimización, CPLEX, DOCPLEX, Programación lineal, Producción industrial

Índice general

1. Introducción	1
1.1. Descripción del problema	1
1.2. Descripción del funcionamiento de la planta	2
1.3. Trabajos relacionados	4
1.4. Cronograma de trabajo	5
1.5. Organización del documento	6
2. Fundamentos teóricos	9
2.1. Programación lineal	9
2.2. Programación lineal entera	10
2.3. CPLEX	10
3. Modelado	13
3.1. Primer modelado	13
3.1.1. Introducción	13
3.1.2. Descripción del funcionamiento de la planta	13
3.1.3. Parámetros	14

3.1.4. Variables	14
3.1.5. Función objetivo	14
3.1.6. Restricciones	15
3.1.7. Conclusiones	15
3.2. Segundo modelado	16
3.2.1. Introducción	16
3.2.2. Parámetros	17
3.2.3. Variables	17
3.2.4. Función objetivo	18
3.2.5. Restricciones	18
3.2.6. Conclusiones	20
3.3. Modelado final	20
3.3.1. Introducción	20
3.3.2. Parámetros	21
3.3.3. Variables	21
3.3.4. Función objetivo	22
3.3.5. Restricciones	22
3.3.6. Conclusiones	24
4. Pruebas de correctitud	25
4.1. Validación de correctitud del primer modelado	26
4.1.1. Parámetros utilizados	26
4.1.2. Prueba 1	26
4.1.3. Prueba 2	27

4.1.4. Prueba 3	28
4.2. Validación de correctitud del segundo modelado	30
4.2.1. Parámetros utilizados	30
4.2.2. Prueba 1	31
4.2.3. Prueba 2	32
4.2.4. Prueba 3	33
4.2.5. Prueba 4	34
4.2.6. Prueba 5	36
4.3. Validación de correctitud del modelado final	37
4.3.1. Parámetros utilizados	37
4.3.2. Prueba 1	38
4.3.3. Prueba 2	40
4.3.4. Prueba 3	41
5. Pruebas de sensibilidad de parámetros	45
5.1. Análisis del rendimiento de las etapas	46
5.2. Análisis de la capacidad de las tolvas	47
5.3. Análisis de la cantidad de horas	49
6. Consideraciones finales	51
6.1. Conclusiones	51
6.2. Desvíos con respecto a la propuesta inicial	52
6.3. Trabajo futuro	53
Referencias	57

A. Código Python del modelo final	59
B. Resultados de las pruebas de correctitud	71
B.1. Pruebas de correctitud del primer modelado	71
B.2. Pruebas de correctitud del segundo modelado	73
B.3. Pruebas de correctitud del modelado final	75

Capítulo 1

Introducción

1.1. Descripción del problema

La cría de ganado es una tarea central en la actividad agropecuaria de Uruguay. Los alimentos balanceados son uno de los principales insumos utilizados en la misma, por lo que su producción impacta fuertemente en muchos de los procesos productivos de la actividad agropecuaria del país. Si bien la cantidad de alimento balanceado demandado fluctúa a lo largo del año, fundamentalmente debido a factores climáticos como el volumen de lluvias, y su impacto sobre la disponibilidad de alimento a partir de pasturas naturales y praderas artificiales, las plantas de raciones suelen trabajar de forma ininterrumpida.

Los alimentos balanceados se suelen expedir en dos distintos formatos: a granel y embolsados; y a su vez en dos distintas presentaciones: harinas y pellets. Cada combinación anterior implica distintos procesos productivos en la planta de elaboración de los alimentos (molienda, dosificado, mezclado, pelletizado, embolsado y expedición) dependiendo del formato y la presentación en cuestión. Por ejemplo, un alimento de presentación harina, expedido a granel, no va a pasar por las etapas de pelletizado ni embolsado; mientras que uno de presentación pelletizado, expedido en bolsas, sí.

Los alimentos balanceados son elaborados a partir de distintas materias primas. Las mismas se pueden dividir en macrocomponentes y microcomponentes de acuerdo a su concentración en el producto final. Los macrocomponentes suelen ser granos, mientras que los microcomponentes son básicamente sales y aditivos. Distintas combinaciones de estos componentes dan lugar a distintos tipos de alimentos. Es importante considerar estos distintos tipos de alimentos en el

modelado, ya que animales de distinta especie conllevan requisitos nutricionales distintos, que van a ser satisfechos por diferentes tipos de alimentos.

Dada una planta, cada una de las etapas tiene una determinada capacidad de producción por hora, y una cantidad de empleados requeridos para que la misma funcione. En general, los empleados trabajan bajo la modalidad de jornaleros, por lo que para cada día se puede decidir la cantidad de empleados que se asignarán a cada etapa. Por lo tanto, resulta interesante optimizar esta planificación de jornales, de forma de minimizar la cantidad de jornales requeridos para satisfacer una determinada demanda.

En este marco, el objetivo del presente trabajo es desarrollar un modelo de programación lineal que represente la realidad descrita anteriormente, y permita optimizar la contratación de jornales para el desarrollo de la actividad productiva. El problema a afrontar es, dada una demanda semanal de distintos tipos de alimentos con distintos tiempos de retiros, minimizar la cantidad de jornales requeridos para que toda la demanda sea satisfecha en tiempo y forma. El modelo deberá considerar restricciones que vienen dadas por el problema planteado, además de las restricciones propias del funcionamiento de la planta. Se espera que, dado una demanda semanal de alimentos, el modelo retorne una planificación detallada, indicando para cada momento las etapas que se encuentran activas, y el tipo de alimento que están produciendo.

1.2. Descripción del funcionamiento de la planta

Como se mencionó en la sección anterior, la elaboración del alimento balanceado se desarrolla en distintas etapas. En el tipo de planta que se considera en este trabajo existen las siguientes etapas: molienda, dosificado, mezclado, pelletizado (solo para pellets), expedición (solo a granel) y embolsado (solo para bolsas/bolsones). A efectos prácticos, como las primeras tres etapas aplican para todos los casos, se las agrupó bajo el nombre “Etapa 0”. Las etapas finales (expedición y embolsado) cuentan con una tolva, depósito de gran tamaño destinado al depósito y canalización de materiales granulares o pulverizados, con el objetivo de permitir el almacenamiento del alimento previo a su uso. La figura 1.1 ilustra las tolvas, las etapas, y cómo se vinculan entre ellas.

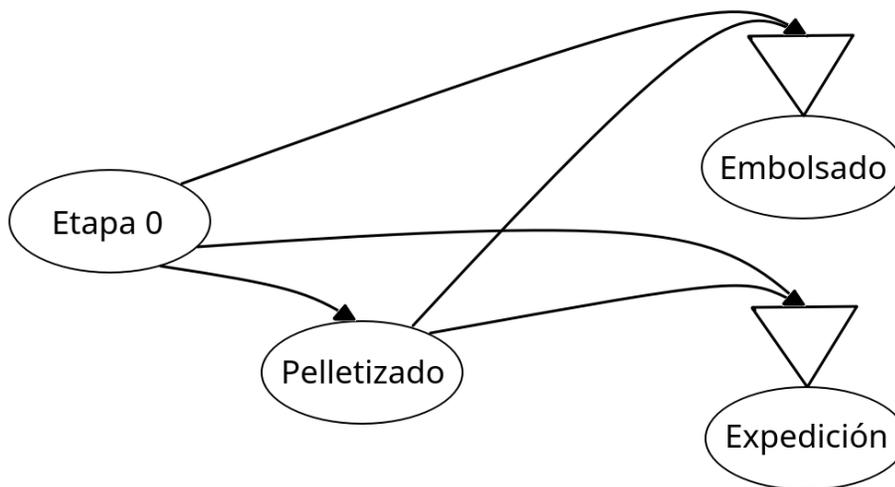


Figura 1.1: Etapas de la planta

Para que cada etapa funcione se requiere de un determinado número de empleados que se encuentren trabajando en la misma. Este número, junto a la cantidad de alimento que procesa cada etapa (toneladas/hora) son parámetros que se necesitan para definir cada etapa. Además, el otro parámetro requerido es la capacidad de cada una de las tolvas.

La planta produce alimentos en dos presentaciones: pellets y harinas, expedidos en dos formatos: embolsados o a granel. Cada demanda de alimento consiste en un identificador que establece el tipo de alimento, una presentación, un formato, una cantidad de alimento (en toneladas), y un día y hora en el que se debe entregar. Esto implica una restricción al modelo, ya que en la definición del problema se estableció que la demanda debe ser satisfecha en su totalidad.

En cada momento, cada etapa de la planta se encuentra produciendo un alimento de cierto tipo, y cada tolva puede estar almacenando alimento de un solo tipo. Cada vez que cambia el tipo de alimento que se está produciendo en la etapa 0, se debe realizar una limpieza para evitar la contaminación del siguiente alimento a producirse. Esta limpieza implica un tiempo muerto de una cierta cantidad minutos en la Etapa 0. Este número de minutos es otro parámetro del modelo.

En cuanto a los empleados, como se mencionó anteriormente, trabajan bajo modalidad de jornaleros, por lo que se tiene libertad de decidir qué empleados asignar a cada etapa en cada día. Se asume que los empleados son especializados, es decir que están capacitados para trabajar en la etapa específica para la cuál se lo asignó, y no en otras.

Cabe destacar que las plantas de producción de alimentos balanceados pueden funcionar bajo distintas configuraciones. El modelo de planta elegido busca ser una opción viable y realista, aunque puede no aplicar exactamente a otros modelos de plantas que siguen otras configuraciones, para los cuales serían necesarios algunos ajustes al modelo.

1.3. Trabajos relacionados

En esta sección se presentan trabajos que de cierta forma se encuentran relacionados con este proyecto. Ya sea porque el tema que tratan está vinculado, o porque por algún motivo, por ejemplo la metodología aplicada, han sido de utilidad para la elaboración del mismo.

En la tesis de Diego Chachapoya sobre producción de alimentos balanceados (Chachapoya, 2014), se presenta el diseño de una planta agroindustrial de producción y comercialización de alimentos balanceados. Si bien el trabajo de Chachapoya no está enfocado en la optimización de la producción del alimento, su trabajo fue de gran ayuda para lograr entender el funcionamiento de las plantas de producción de alimentos balanceados, lo cual fue un insumo importante a la hora de definir el modelado del problema. Además de presentar los distintos procesos y equipos involucrados en la producción del alimento, el autor da un panorama general sobre los distintos tipos de alimentos que se pueden producir y las materias primas necesarias para la tarea.

En el artículo de Valencia, Aguirre y Herrera (Valencia, Aguirre, y Herrera, 2018) los autores exponen una revisión de la literatura relativa al tema de programación de recursos y planificación de actividades en la agricultura, haciendo foco en las distintas herramientas de optimización empleadas en los trabajos relevados. El documento presenta las razones por las cuales es de especial interés realizar dicha revisión sistemática de la literatura, y explica la metodología utilizada para el relevamiento bibliográfico. Como resultado de su investigación, los autores encontraron 52 artículos relevantes de un total de 692 artículos analizados. Para cada uno de estos artículos se presenta un breve resumen detallando los objetivos del mismo. Finalmente, los autores realizan una discusión de los resultados, comparando los distintos enfoques que se tomaron en cada uno de los artículos. Este trabajo fue de gran utilidad a la hora de tener una idea general de los distintos enfoques y problemas que se han abordado en el campo de la agricultura, muchos de los cuales tienen ciertos puntos de contacto con el trabajo aquí presentado.

En el trabajo de Richard Vargas y Jaime Eduardo Gutiérrez sobre aplicaciones de la programación lineal para optimizar costos de dietas balanceadas (Vargas y Gutierrez, 2018), los autores se centran en el uso de la programación

lineal para minimizar el costo de la dieta de alimento balanceado para pollos en la industria avícola. Para poder determinar las restricciones que se le iban a imponer a las variables de decisión, los autores debieron recabar información sobre los componentes nutricionales requeridos por los animales. Luego, se enfocaron en encontrar la mezcla óptima de insumos que cumple con los requerimientos nutricionales de los animales. Aunque el problema que se plantearon Vargas y Gutiérrez difiere del que se trata en este proyecto, su trabajo sirvió como un primer acercamiento a los procesos de optimización que se pueden conseguir aplicando programación lineal en el campo de la producción de alimentos balanceados.

En el trabajo de Marcela Castillo y José Carvajal ([Castillo y Carvajal, 2019](#)) se trata el problema de asignación de horarios de instructores en un centro educativo de forma de optimizar la capacidad instalada en el centro, maximizando la utilización de los recursos disponibles, teniendo en cuenta la demanda de cada programa de formación. Si bien a priori puede parecer que el problema abordado por Castillo y Carvajal no guarda demasiada similitud con el tratado en este trabajo, la realidad es que se puede trazar cierto paralelismo entre los instructores y los operarios de la planta de alimentos, entre los programas de formación y los alimentos balanceados, y entre los recursos disponibles en el centro y las distintas etapas del proceso de producción del alimento balanceado. De esta forma, ambos problemas de asignación de recursos comparten parte de su estructura, y sin duda algunas de las decisiones tomadas por los autores sirvieron de inspiración para este trabajo.

1.4. Cronograma de trabajo

A la hora de comenzar con el proyecto se definió un cronograma de trabajo tentativo. El mismo indicaba distintas tareas que debían realizarse, junto al tiempo estimado que se les dedicaría. En la tabla [1.1](#) se muestran las distintas tareas junto al tiempo estimado de cada una. El cronograma sirvió de referencia a la hora de planificar las distintas etapas del proyecto, además de definir una hoja de ruta la cual seguir. En las conclusiones del documento se comentan los resultados del cronograma. Es decir, la cantidad de tiempo que efectivamente se le dedicó a cada tarea, junto a posibles motivos que pueden explicar dichos desvíos.

Tarea	Duración estimada
Relevamiento bibliográfico y armado del estado del arte con problemas relacionados.	2 meses
Modelización e Implementación del Problema.	1.5 meses
Testeo del Modelo desarrollado usando herramientas de optimización como ser CPLEX.	1.5 meses
Estudio de sensibilidad de la solución frente a variaciones en los parámetros. Analizar distintas combinaciones de parámetros y sus resultados	1 mes
Documentación del proyecto.	2 meses

Tabla 1.1: Cronograma estimado de trabajo

1.5. Organización del documento

El documento está organizado en capítulos, y a su vez cada capítulo se divide en secciones.

En el [capítulo 1](#) se presenta el problema. Se da contexto sobre la temática de alimentos balanceados, y se describe el modelo de planta sobre el que se basa el trabajo. Además se presentan distintos trabajos relacionados que han servido como referencia.

Luego, el [capítulo 2](#) presenta los fundamentos teóricos en los que se basa el trabajo. Además describe algunas de las herramientas que fueron utilizadas para la elaboración del mismo.

Una vez introducido el problema y los fundamentos teóricos, en el [capítulo 3](#) se presenta el modelado matemático. El modelado fue desarrollado de forma incremental, partiendo de un modelo básico, el cual se fue complejizando hasta alcanzar un modelo lo suficientemente realista. Este proceso incremental de modelado se dividió en tres fases. Para cada una de estas fases existe una sección en el capítulo, las cuales presentan los objetivos de la fase, el modelado que se construyó, y las conclusiones o resultados de la fase.

El [capítulo 4](#) está dedicado a las pruebas de correctitud de los modelados. Las pruebas de correctitud tienen el objetivo de validar que cada uno de los modelados tiene el comportamiento esperado. Es decir, que los resultados que brindan son consistentes con las restricciones y objetivo de cada uno.

En el [capítulo 5](#) se presentan las pruebas de sensibilidad de parámetros. Estas

pruebas se realizaron solamente sobre el modelado final, y tienen la finalidad de analizar el impacto de cada uno de los parámetros sobre el modelo. Las pruebas consisten en realizar variaciones de los valores de ciertos parámetros, y evaluar cómo afectan estas variaciones a los resultados brindados por el modelado.

Finalmente, el [capítulo 6](#) presenta las conclusiones del trabajo realizado. También se proponen distintas ideas para trabajos futuros que no entraron en el alcance de este proyecto, que podrían abordarse en un eventual trabajo de maestría por ejemplo.

Capítulo 2

Fundamentos teóricos

2.1. Programación lineal

La programación lineal es un campo de la matemática dedicado a optimizar sistemas lineales. En concreto, se busca maximizar o minimizar una función lineal, denominada función objetivo, sujeta a restricciones lineales. Estas restricciones se suelen expresar mediante un sistema de ecuaciones e inecuaciones lineales. En general se exige también que las variables sean positivas. Un problema general de programación lineal se puede escribir de la siguiente forma:

$$\left\{ \begin{array}{ll} \min & z = \sum_{j=1}^n c_j x_j \\ \text{sujeto a} & \sum_{j=1}^n a_j x_j \geq b_i, \quad i = 1, \dots, m \\ & x_j \geq 0 \quad j = 1, \dots, n \end{array} \right.$$

La programación lineal es ampliamente utilizada para resolver problemas de planificación, asignación de recursos y toma de decisiones. En general, el método utilizado para resolver estos problemas es el Método Simplex. Este es un método iterativo, que recorre los vértices adyacentes del poliedro que forma la región factible hasta encontrar la solución óptima.

2.2. Programación lineal entera

Los modelos de programación lineal entera son modelos de programación lineal en los cuales se agrega la restricción de que al menos alguna variable de decisión debe ser entera. En muchos casos la solución del programa lineal truncada (o redondeada) está lejos de ser el óptimo para este tipo de problemas, por lo que surgen distintos métodos para hallar una solución exacta a este tipo de problemas.

Uno de los métodos más utilizados para resolver estos problemas es el Método de ramificación y acotación (branch and bound). Este método comienza resolviendo el problema lineal asociado (PLA), que consiste en eliminar las condiciones de integralidad, por ejemplo con el método Simplex. Si esta solución no satisface las condiciones de integralidad, se comienza con la etapa de ramificación.

La ramificación consiste en dividir el problema en dos nuevos problemas eliminando la restricción de integralidad. Por ejemplo, si la variable x_i debía ser entera, y toma el valor x_{bi} no entero, se generan las siguientes dos restricciones: $x_i \leq \lfloor x_{bi} \rfloor$ y $x_i \geq \lceil x_{bi} \rceil$. Este proceso se repite hasta que todos los subproblemas tengan solución entera o no factible.

Utilizando solo la ramificación, la cantidad de problemas crece exponencialmente. Por este motivo, la ramificación se combina con la acotación. La acotación se basa en que la solución óptima de cada subproblema va a ser peor que la solución óptima del problema padre. Entonces, la acotación consiste en tomar como cota inferior (o superior en caso de problemas de minimización) aquella solución entera con el mejor valor obtenido hasta el momento, y descartar cualquier subproblema con resultado óptimo peor que el dado por la cota.

2.3. CPLEX

CPLEX (IBM, 2023) es un paquete de software desarrollado por IBM orientado a la resolución de problemas de optimización. La herramienta permite resolver diversos problemas, entre ellos, problemas de programación lineal y programación lineal entera. Para los problemas de programación lineal entera es capaz de utilizar distintos métodos, entre ellos, el de ramificación y acotación.

CPLEX proporciona una capa de modelado con interfaces para distintos lenguajes de programación. El modelo elaborado para resolver el problema presentado anteriormente utiliza la interfaz para el lenguaje *Python* (IBM, 2021) (“IBM Decision Optimization CPLEX Modeling for Python”, 2022). Esta in-

terfaz está basada en la API para el lenguaje *C*, y ofrece prácticamente todas las funcionalidades de CPLEX. Los modelos elaborados pueden ser resueltos localmente (con la misma API para *Python*) o en la nube.

DOcplex (la API de CPLEX para Python) permite definir el modelo, con su función objetivo y restricciones en Python. A la vez, brinda diversas facilidades a la hora de describir el modelado. Por ejemplo, permite definir restricciones que involucran condiciones lógicas del estilo $A \Rightarrow B$ de forma directa. Ya que la herramienta lo permite, en la sección del modelado algunas restricciones se dejaron expresadas de esta forma.

Capítulo 3

Modelado

Este capítulo está dedicado al modelado matemático del problema descrito anteriormente. El mismo fue desarrollado iterativamente, comenzando con un modelado muy básico, e incorporando de a poco los elementos necesarios para satisfacer el modelo de planta propuesto. De esta forma, se pudo ir validando el modelado a medida que era construido, sin necesidad de llegar al modelado final para hacer los ajustes requeridos.

3.1. Primer modelado

3.1.1. Introducción

Este modelado es una primera aproximación al problema, por lo que lo mencionado en la sección 1.2 no aplica en su totalidad para esta primera versión. El propósito de este modelado es tener un primer prototipo funcional, que considere una versión simplificada de la realidad planteada anteriormente. En esta instancia, la función objetivo a minimizar es el número de horas de actividad de la planta requeridas para satisfacer la demanda.

3.1.2. Descripción del funcionamiento de la planta

El modelo de planta considerado no tiene un número fijo de etapas, y produce alimentos en un único formato y un único tipo de presentación. Para la elaboración del alimento, el mismo debe pasar por cada una de las etapas de

forma ordenada. La demanda, que debe ser satisfecha en su totalidad, consiste en un identificador del alimento, una cantidad (en toneladas), un día y una hora.

3.1.3. Parámetros

- D : Cantidad de días disponibles para la producción
- H : Cantidad de horas por día disponibles para la producción
- I : Cantidad de etapas
- Cap_i : Rendimiento de la etapa i (en toneladas por hora)
- $Dem_{a,h,d}$: Cantidad (en toneladas) de alimento con identificador a que se debe entregar a la hora h del día d

3.1.4. Variables

Variables de decisión

- $prod_{a,i,h,d} \in \mathbb{R}$: Cantidad de alimento a producido en la etapa i en la hora h del día d

Variables Auxiliares

- $activa_{h,d} \in \{0, 1\}$: Vale 1 si la planta está activa a la hora h del día d , y 0 en caso contrario

3.1.5. Función objetivo

Se busca minimizar la cantidad de horas en que la planta se encuentra activa.

$$\text{mín} \sum_{h=1}^H \sum_{d=1}^D activa_{h,d}$$

3.1.6. Restricciones

- **Demanda satisfecha:**

$$\sum_{h=1}^H \sum_{d=1}^{d'-1} prod_{a,I,h,d} + \sum_{h=1}^{h'} prod_{a,I,h,d'} \geq dem_{a,h',d'} ,$$

$$\forall dem_{a,h',d'}$$

Suma de lo producido en días anteriores, más lo producido hasta la hora de entrega del mismo día es mayor o igual a lo demandado. Donde I es la última etapa (la cual representa la entrega del alimento producido).

- **Capacidad de las etapas no desbordada:**

$$prod_{a,i,h,d} \leq Cap_i, \forall a; i = 1, \dots, I; h = 1, \dots, H; d = 1, \dots, D$$

En ningún momento lo producido en una etapa supera su capacidad de producción.

- **Integridad entre etapas:**

$$prod_{a,i,h,d} \leq \sum_{h'=1}^H \sum_{d'=1}^{d-1} prod_{a,i-1,h',d'} + \sum_{h'=1}^h prod_{a,i-1,h',d} -$$

$$- \sum_{h'=1}^H \sum_{d'=1}^{d-1} prod_{a,i,h',d'} - \sum_{h'=1}^{h-1} prod_{a,i,h',d}$$

$$\forall a; i = 2, \dots, I; h = 1, \dots, H; d = 1, \dots, D$$

En cada momento, cada etapa no puede utilizar más alimento del que tiene disponible. Es decir, la etapa i no puede producir más de lo que produjo la etapa $i - 1$ menos lo que ya produjo la etapa i hasta el momento.

- **Planta activa:**

$$\sum_a \sum_{i=1}^I prod_{a,i,h,d} > 0 \Rightarrow activa_{h,d} = 1, h = 1, \dots, H; d = 1, \dots, D$$

En todo momento, si alguna etapa de la planta se encuentra activa, entonces la variable *activa* debe valer 1.

3.1.7. Conclusiones

El primer modelado sirvió como un acercamiento inicial al problema, y sentó las bases de los siguientes modelados. A pesar de que la realidad planteada en esta instancia está lejos de ser un fiel reflejo del funcionamiento real de una

planta de producción de alimentos balanceados, se logró un prototipo funcional, que brinda resultados óptimos en este contexto simplificado.

Además de ser un primer contacto con el funcionamiento de las plantas de producción de alimentos, este modelado sirvió para habituarse a las tecnologías que fueron usadas a lo largo del proyecto. En esta etapa se definió el uso de la API de CPLEX para Python, y se elaboró el primer Jupyter Notebook ([Jupyter, 2023](#)) con la estructura de código que se usaría a lo largo del proyecto. También se probaron distintas herramientas para la visualización de datos de las soluciones brindadas por el algoritmo, evaluando alternativas que permitieran plasmar la mayor cantidad de información posible.

Aunque quizás parezca que el problema se simplificó en exceso para esta instancia, todas estas valoraciones de las distintas alternativas para la implementación del modelado hubieran sido difíciles de realizar en un modelo más complejo que reflejara con mayor precisión el real funcionamiento de la planta.

3.2. Segundo modelado

3.2.1. Introducción

Una vez cumplidos los objetivos propuestos en el primer modelado, se procedió a idear un modelado que incorporara los componentes necesarios para lograr una representación más precisa de la realidad. El segundo modelado agregó el concepto de los distintos formatos y presentaciones de los alimentos (harinas/pelletizados y granel/bolsas). De esta forma, cada demanda de alimento consiste en el producto demandado (formato + presentación), una cantidad, y un día y hora en que tiene que ser entregada. Además, se fijaron las etapas correspondientes (etapa 0, pelletizado, expedición y embolsado), y se definieron tolvas para las etapas de expedición y embolsado.

Nótese que, si bien se incorporaron al modelado los distintos formatos y presentaciones de los alimentos, la planta sigue produciendo un único tipo de alimento. Esto significa que la composición de los alimentos es única, solamente distinguiéndose por su formato y presentación.

A efectos prácticos, se dividió cada una de las etapas de expedición y embolsado en dos subetapas correspondientes a cada tipo de presentación. Es decir, en lugar de tener una etapa de expedición y una de embolsado, se tiene una etapa de expedición en pellets, una de expedición de harinas, una de embolsado de pellets y una de embolsado de harinas. Además, para cada una de estas etapas adicionales, se agregó una etapa final, que simboliza que el alimento ya fue

entregado. Esta última etapa es necesaria, ya que al no tener un identificador de cada alimento (porque son todos del mismo tipo), se requiere una forma de poder distinguir entre los alimentos que ya han sido entregados y los que no.

En este modelado se cambió la función objetivo a optimizar. En vez de minimizar la cantidad de horas de actividad de la planta requeridas para satisfacer la demanda, se pasó a minimizar el número de jornales requeridos para satisfacerla. El cambio en la función objetivo se da porque se entendió que el análisis de los resultados podía ser más rico de esta forma. De todas formas, revertir este cambio, y volver a minimizar la cantidad de horas de actividad requeriría un simple ajuste al modelo.

3.2.2. Parámetros

- D : Cantidad de días disponibles para la producción
- H : Cantidad de horas por día disponibles para la producción
- Cap_i : Rendimiento de la etapa i (en toneladas por hora)
- Emp_i : Cantidad de empleados que deben estar trabajando en la etapa i para que la misma funcione
- $CapAlm_i$: Capacidad (en toneladas) de almacenamiento de la tolva correspondiente a la etapa i (expedición o embolsado)
- $Dem_{a,h,d}$: Cantidad (en toneladas) de alimento a que se debe entregar a la hora h del día d , donde “ a ” permite discernir el formato y la presentación del alimento.

3.2.3. Variables

Variables de decisión

- $prod_{i,h,d} \in \mathbb{R}$: Cantidad de alimento (en toneladas) producido en la etapa i en la hora h del día d
- $alm_{i,h,d} \in \mathbb{R}$: Indica la cantidad de alimento (en toneladas) que ingresó a la tolva correspondiente a la etapa i (expedición o embolsado) a la hora h del día d

Variables Auxiliares

- $emp_act_{i,d} \in \mathbb{N}$: Indica la cantidad de empleados que se encuentran trabajando en la etapa i en el día d

3.2.4. Función objetivo

Se busca minimizar el número de jornales (empleados) requeridos para satisfacer la demanda.

$$\text{mín} \sum_i^{Etapas} \sum_{d=1}^D emp_act_{i,d}$$

3.2.5. Restricciones

A modo de simplificar la escritura de las restricciones, se utilizará el siguiente atajo para escribir la cantidad de alimento disponible en la tolva correspondiente a la etapa i en la hora h del día d :

$$\begin{aligned} alim_disp_{i,h,d} = & \sum_{h'=1}^H \sum_{d'=1}^{d-1} alm_{i,h',d'} + \sum_{h'=1}^h alm_{i,h',d} - \\ & \sum_{h'=1}^H \sum_{d'=1}^{D-1} prod_{i,h',d'} - \sum_{h'=1}^{h-1} prod_{i,h',d} \end{aligned}$$

- **Demanda satisfecha:**

$$\sum_{h=1}^H \sum_{d=1}^{d'-1} prod_{i_f,h,d} + \sum_{h=1}^{h'} prod_{i_f,h,d'} \geq dem_{a,h',d'} ,$$

$$\forall dem_{a,h',d'}$$

Suma de lo producido en días anteriores, más lo producido hasta la hora de entrega del mismo día es mayor o igual a lo demandado. Donde i_f representa a la etapa final (de entrega) correspondiente al alimento a (según su formato y presentación).

- **Capacidad de las etapas no desbordada:**

$$prod_{i,h,d} \leq Cap_i, \forall i; h = 1, \dots, H; d = 1, \dots, D$$

En ningún momento lo producido en una etapa supera su capacidad de producción.

■ **Capacidad de las tolvas no desbordada:**

$$alim_disp_{i,h,d} \leq CapAlm_i, \quad \forall i; h = 1, \dots, H; d = 1, \dots, D$$

Ninguna tolva puede almacenar más alimento del que indica su capacidad.

■ **Capacidad de producción no desbordada:**

$$prod_{i,h,d} \leq alim_disp_{i,h,d}, \quad \forall i; h = 1, \dots, H; d = 1, \dots, D$$

En ningún momento se puede producir (expedir o embolsar) más alimento del que se encuentra en la tolva correspondiente.

■ **Flujo:**

$$prod_{etapa.0,h,d} = prod_{pellet,h,d} + prod_{exp.har,h,d} + prod_{emb.har,h,d}, \\ h = 1, \dots, H; d = 1, \dots, D$$

En todo momento, lo producido en la etapa 0 es igual a la suma de lo producido en la etapa de pelletizado, la etapa de expedición de harinas y la etapa de embolsado de harinas.

■ **Flujo pelletizado:**

$$prod_{pellet,h,d} = alm_{exp.pell,h,d} + prod_{emb.pell,h,d}, \\ h = 1, \dots, H; d = 1, \dots, D$$

En todo momento, lo producido en la etapa de pelletizado es igual a la suma de lo producido en la etapa de embolsado de pelletizado y la etapa de expedición de pelletizado.

■ **Mutex harina-pelletizado:**

$\forall h = 1, \dots, H; d = 1, \dots, D :$

- $alim_disp_{exp.har,h,d} \times alim_disp_{exp.pell,h,d} = 0$
- $alim_disp_{emb.har,h,d} \times alim_disp_{emb.pell,h,d} = 0$

Al haber una única tolva por etapa de expedición y embolsado, en cada momento, cada tolva puede estar almacenando alimento de un solo tipo (harina o pellets). Esta restricción es necesaria debido a la división en subetapas de las etapas de expedición y embolsado.

■ **Empleados activos:**

$\forall d = 1, \dots, D :$

- $\sum_{h=1}^H prod_{etapa.0,h,d} > 0 \Rightarrow emp_act_{etapa.0,d} = Emp_{etapa.0}$
- $\sum_{h=1}^H prod_{pellet,h,d} > 0 \Rightarrow emp_act_{pellet,d} = Emp_{pellet}$

- $\sum_{h=1}^H prod_{exp_har,h,d} + prod_{exp_pell,h,d} > 0 \Rightarrow emp_act_{exp,d} = Emp_{exp}$
- $\sum_{h=1}^H prod_{emb_har,h,d} + prod_{emb_pell,h,d} > 0 \Rightarrow emp_act_{emb,d} = Emp_{emb}$

Si en algún momento del día una determinada etapa se encuentra activa, entonces deben estar los correspondientes empleados activos. El caso de las etapas de expedición y embolsado es especial, ya que están separadas en subetapas simplemente a efectos prácticos del modelado, pero en la realidad se corresponden a una única etapa.

3.2.6. Conclusiones

Este segundo modelado pudo incorporar muchos de los aspectos que en un principio se consideraron fundamentales para el problema. Se logró plasmar de forma mucho más realista el funcionamiento de la planta, y sirvió para validar que se estaba yendo en la dirección correcta.

A medida que se fue ideando e implementando, dejó ver algunos aspectos a mejorar, los cuales se buscó incorporar en el modelado final. Por ejemplo, agregar la posibilidad de producir distintos tipos de alimentos (mas allá de su presentación y formato), y de la mano de esto la ya mencionada penalización por cambio de alimento producido.

3.3. Modelado final

3.3.1. Introducción

El modelado final está fuertemente basado en el segundo modelado: las etapas son las mismas, la función objetivo es la misma, y el funcionamiento y estructura general de la planta se mantienen iguales. Como se señaló anteriormente, durante la implementación del segundo modelado se detectaron algunos puntos débiles, que no terminaban de reflejar bien la realidad de la producción de alimentos balanceados, aspectos que se procuró corregir en esta formulación final del modelado.

Una clara deficiencia del modelado anterior era la falta de variedad de alimentos. Las plantas de alimentos balanceados pueden ofrecer distintos tipos de alimentos, que se elaboran a partir de distintas combinaciones de macro y micro componentes. Por esto, en el nuevo modelado la planta produce distintos

tipos de alimentos. De esta forma, cada demanda de alimento se conforma por un formato, una presentación y, además, una determinada composición correspondiente a este alimento. Cada demanda se corresponde con un identificador único que permite determinar todas las propiedades del alimento demandado (formato, presentación y composición).

De la mano de la incorporación de los distintos tipos de alimentos al modelado, se agregó también una penalización por el cambio del alimento a producir en la etapa 0. Esta penalización es de 30 minutos, en los cuales la etapa 0 debe mantenerse sin producir. La justificación de esta penalización viene dada por la necesidad de limpiar ciertos componentes, como las mezcladoras, una vez que se cambia el alimento con el que estaban trabajando. Para soportar estas pausas de 30 minutos, se debió cambiar la granularidad utilizada para representar el tiempo. Hasta el momento el tiempo era representado por días y horas, lo cual no es suficiente para expresar interrupciones de 30 minutos, por lo que se debió pasar a una representación del tiempo en días y medias horas.

3.3.2. Parámetros

- D : Cantidad de días disponibles para la producción
- H : Cantidad de horas por día disponibles para la producción
- Cap_i : Rendimiento de la etapa i (en toneladas por hora)
- Emp_i : Cantidad de empleados que deben estar trabajando en la etapa i para que la misma funcione
- $CapAlm_i$: Capacidad (en toneladas) de almacenamiento de la tolva correspondiente a la etapa i (expedición o embolsado)
- $Dem_{a,h,d}$: Cantidad (en toneladas) de alimento a que se debe entregar a la hora h del día d

3.3.3. Variables

Variables de decisión

- $prod_{a,i,h,d} \in \mathbb{R}$: Cantidad de alimento a (en toneladas) producido en la etapa i en la (media) hora h del día d
- $alm_{a,i,h,d} \in \mathbb{R}$: Indica la cantidad de alimento a (en toneladas) que ingresó a la tolva correspondiente a la etapa i a la (media) hora h del día d

Variables Auxiliares

- $emp_act_{i,d} \in \mathbb{N}$: Indica la cantidad de empleados que se encuentran trabajando en la etapa i en el día d
- $prod_bool_{a,i,h,d} \in \{0,1\}$: Vale 1 si se está produciendo el alimento a en la etapa i a la (media) hora h del día d , y 0 si no

3.3.4. Función objetivo

Se busca minimizar el número de jornales (empleados) requeridos para satisfacer la demanda.

$$\text{mín} \sum_i^{Etapas} \sum_{d=1}^D emp_act_{i,d}$$

3.3.5. Restricciones

A modo de simplificar la escritura de las restricciones, se utilizará el siguiente atajo para escribir la cantidad de alimento a disponible en la tolva correspondiente a la etapa i en la (media) hora h del día d :

$$alim_disp_{a,i,h,d} = \sum_{h'=1}^{2H} \sum_{d'=1}^{d-1} alm_{a,i,h',d'} + \sum_{h'=1}^h alm_{a,i,h',d} - \sum_{h'=1}^{2H} \sum_{d'=1}^{d-1} prod_{a,i,h',d'} - \sum_{h'=1}^{h-1} prod_{a,i,h',d}$$

- **Demanda satisfecha:**

$$\sum_{h=1}^{2H} \sum_{d=1}^{d'-1} prod_{a,i_f,h,d} + \sum_{h=1}^{2h'} prod_{a,i_f,h,d'} \geq dem_{a,h',d'} ,$$

$$\forall dem_{a,h',d'}$$

Suma de lo producido en días anteriores, más lo producido hasta la hora de entrega del mismo día es mayor o igual a lo demandado. Donde i_f representa a la última etapa correspondiente al alimento a (según su formato y presentación).

■ **Capacidad de las etapas no desbordada:**

$$prod_{a,i,h,d} \leq prod_bool_{a,i,h,d} \times \frac{Cap_i}{2}, \quad \forall a; \forall i; h = 1, \dots, 2H; d = 1, \dots, D$$

En ningún momento lo producido en una etapa supera su capacidad de producción. Se agrega la variable *prod_bool* para forzar que tenga que valer 1 en caso de que la planta se encuentre produciendo. Como la granularidad es de media hora, se considera la mitad de capacidad de producción.

■ **Capacidad de las tolvas no desbordada:**

$$alim_disp_{a,i,h,d} \leq CapAlm_i, \quad \forall a; \forall i; h = 1, \dots, 2H; d = 1, \dots, D$$

Ninguna tolva puede almacenar más alimento del que indica su capacidad.

■ **Capacidad de producción no desbordada:**

$$prod_{a,i,h,d} \leq alim_disp_{a,i,h,d}, \quad \forall a; \forall i; h = 1, \dots, 2H; d = 1, \dots, D$$

En ningún momento se puede producir (expedir o embolsar) más alimento del que se encuentra en la tolva correspondiente.

■ **Flujo:**

$$prod_{a,etapa.0,h,d} = prod_{a,pellet,h,d} + alm_{a,exp.har,h,d} + alm_{a,emb.har,h,d}, \\ \forall a; h = 1, \dots, 2H; d = 1, \dots, D$$

En todo momento, lo producido en la etapa 0 es igual a la suma de lo producido en la etapa de pelletizado, la etapa de expedición de harinas y la etapa de embolsado de harinas.

■ **Flujo pelletizado:**

$$prod_{a,pellet,h,d} = alm_{a,exp.pell,h,d} + alm_{a,emb.pell,h,d}, \\ \forall a; h = 1, \dots, 2H; d = 1, \dots, D$$

En todo momento, lo producido en la etapa 0 es igual a la suma de lo producido en la etapa de pelletizado, la etapa de expedición de harinas y la etapa de embolsado de harinas.

■ **Mutex harina-pelletizado:**

$$\forall a_1, a_2; h = 1, \dots, 2H; d = 1, \dots, D :$$

- $alim_disp_{a_1,exp.har,h,d} \times alim_disp_{a_2,exp.pell,h,d} = 0$
- $alim_disp_{a_1,emb.har,h,d} \times alim_disp_{a_2,emb.pell,h,d} = 0$

Al haber una única tolva por etapa de expedición y embolsado, en cada momento, cada tolva puede estar almacenando alimento de un solo tipo (harina o pellets).

■ **Mutex entre alimentos:**

$$\sum_{a \in Alim} prod_bool_{a,i,h,d} \leq 1, \quad \forall i; h = 1, \dots, 2H; d = 1, \dots, D$$

En cada momento, cada etapa puede producir un único tipo de alimento.

■ **Penalización por cambio de alimento:**

$$\forall a_1 \neq a_2; h = 1, \dots, 2H - 1; d = 1, \dots, D :$$

$$prod_bool_{a_1,etap_0,h,d} + prod_bool_{a_2,etap_0,h+1,d} \leq 1$$

Para cualquier par de alimentos, la etapa 0 no puede producir en (medias) horas consecutivas ambos alimentos.

■ **Empleados activos:**

$$\forall d = 1, \dots, D :$$

$$\bullet \sum_{a \in Alim} \sum_{h=1}^{2H} prod_{a,etapa_0,h,d} > 0 \Rightarrow emp_act_{etapa_0,d} = Emp_{etapa_0}$$

$$\bullet \sum_{a \in Alim} \sum_{h=1}^{2H} prod_{a,pellet,h,d} > 0 \Rightarrow emp_act_{pellet,d} = Emp_{pellet}$$

$$\bullet \sum_{a \in Alim} \sum_{h=1}^{2H} prod_{a,exp_har,h,d} + prod_{a,exp_pell,h,d} > 0 \Rightarrow emp_act_{exp,d} = Emp_{exp}$$

$$\bullet \sum_{a \in Alim} \sum_{h=1}^{2H} prod_{a,emb_har,h,d} + prod_{a,emb_pell,h,d} > 0 \Rightarrow emp_act_{emb,d} = Emp_{emb}$$

Si en algún momento del día una determinada etapa se encuentra activa, entonces deben estar los correspondientes empleados activos. El caso de las etapas de expedición y embolsado es especial, ya que están separadas en subetapas simplemente a efectos prácticos del modelado, pero en la realidad se corresponden a una única etapa.

3.3.6. Conclusiones

El tercer modelado logró colmar las expectativas que se tenían en cuanto a la representación de la realidad. Por lo tanto, dado por cerrada la etapa de modelado, se procedió a idear un plan de pruebas que permita verificar que el modelo se comporta de la forma esperada ante distintas entradas.

Capítulo 4

Pruebas de correctitud

Para confirmar la correctitud de los modelados se realizaron una serie de pruebas orientadas a verificar que los mismos se comportaran de la forma esperada. Se realizaron pruebas para cada uno de los modelados, comenzando con pruebas sencillas, para luego ir aumentando la complejidad de las mismas.

Para cada modelado se definen previamente los parámetros utilizados en las pruebas. Lo que varía entre las distintas pruebas es la demanda de alimentos, lo cual da lugar a distintas soluciones brindadas por el modelo.

En la mayoría de las pruebas se considerarán demandas para los primeros dos días. De esta forma el análisis será mucho más claro que si se consideraran demandas para cualquier día de la semana. Además, como la mayoría de las pruebas tienen un objetivo concreto, no es necesario extender la producción en más días. De todas formas, al final de esta sección se encuentra una prueba que contempla una demanda semanal completa, de forma de poder validar el modelo final ante un caso realista que bien podría ser un escenario de planificación usual.

Además de los resultados y análisis de las soluciones brindadas por el modelo, para las pruebas del modelado final se incluyen datos de cómo el modelo escala para las distintas instancias del problema. Se incluyen datos del número de variables y tiempo de resolución de la instancia. Todas las pruebas fueron ejecutadas en un computador portátil con 12 GB de RAM, procesador Intel Core i7-8550U, y sistema operativo Linux (Ubuntu 20.04.6)

4.1. Validación de correctitud del primer modelado

4.1.1. Parámetros utilizados

- $D = 5$ (Cantidad de días disponibles para la producción)
- $H = 7$ (Cantidad de horas por día disponibles para la producción)
- $I = 5$ (Cantidad de etapas. Simbolizan: Batcheo, Molienda, Dosificado, Mezclado y Expedición)
- Cap_i (Rendimiento de la etapa i en toneladas por hora)
 - $Cap_1 = 30$
 - $Cap_2 = 12$
 - $Cap_3 = 12$
 - $Cap_4 = 12$
 - $Cap_5 = 15$

4.1.2. Prueba 1

Descripción: Caso sencillo, con una única demanda de poco alimento.

Objetivo: Probar el funcionamiento básico del modelo en el caso más sencillo posible.

Demanda:

Id de alimento	Hora de entrega	Día de entrega	Cantidad
1	7	1	10

Tabla 4.1: Demanda prueba 1 - primer modelado.

Resultados: Solución óptima: 5 horas de actividad.

Como era de esperar, la solución brindada por el modelo simplemente produce el alimento secuencialmente por las distintas etapas. Como el alimento debe pasar por cada una de las etapas, la planta debe estar activa como mínimo por 5 horas, que es el número de etapas. La tabla 4.2 muestra la solución explícita proporcionada por el modelo, es decir la planificación detallada de la producción.

Alimento	Etapa	Hora	Día	Producción
1	1	3	1	10
1	2	4	1	10
1	3	5	1	10
1	4	6	1	10
1	5	7	1	10

Tabla 4.2: Resultados prueba 1 - primer modelado (omitiendo valores nulos).

4.1.3. Prueba 2

Descripción: Similar al caso anterior pero con mayor cantidad de alimento demandado.

Objetivo: Probar el modelo ante una demanda mayor de alimento. Se busca evaluar que la integridad entre las etapas se satisfaga en cada momento, a pesar de que más de una etapa se encuentre activa en determinadas horas.

Demanda:

Id de alimento	Hora de entrega	Día de entrega	Cantidad
1	4	2	80

Tabla 4.3: Demanda prueba 2 - primer modelado.

Resultados: Solución óptima: 11 horas de actividad

Para estos casos en que la solución está compuesta por muchos valores, se presentará la misma mediante gráficos, ya que hacerlo en una tabla no resulta práctico. Para el caso del primer modelado, se mostrará una gráfica para cada etapa indicando la cantidad de alimento que se produce en cada momento. En caso de querer consultar los resultados explícitos, los mismos se encuentran disponibles en el [anexo](#).

Nuevamente, la solución brindada por el modelo produce el alimento secuencialmente por etapas, de forma tal que la restricción de integridad entre las mismas se satisface. En este caso, como la demanda se encuentra próxima a la capacidad máxima de producción de la planta, la misma debe estar activa durante todo el tiempo previo a la entrega del alimento. El modelo no fue capaz de encontrar una planificación que logre satisfacer la demanda con tiempos de parada simultáneos en todas las etapas, lo cual hubiera resultado en una mejor solución. La figura 4.1 muestra la solución brindada por el modelo.

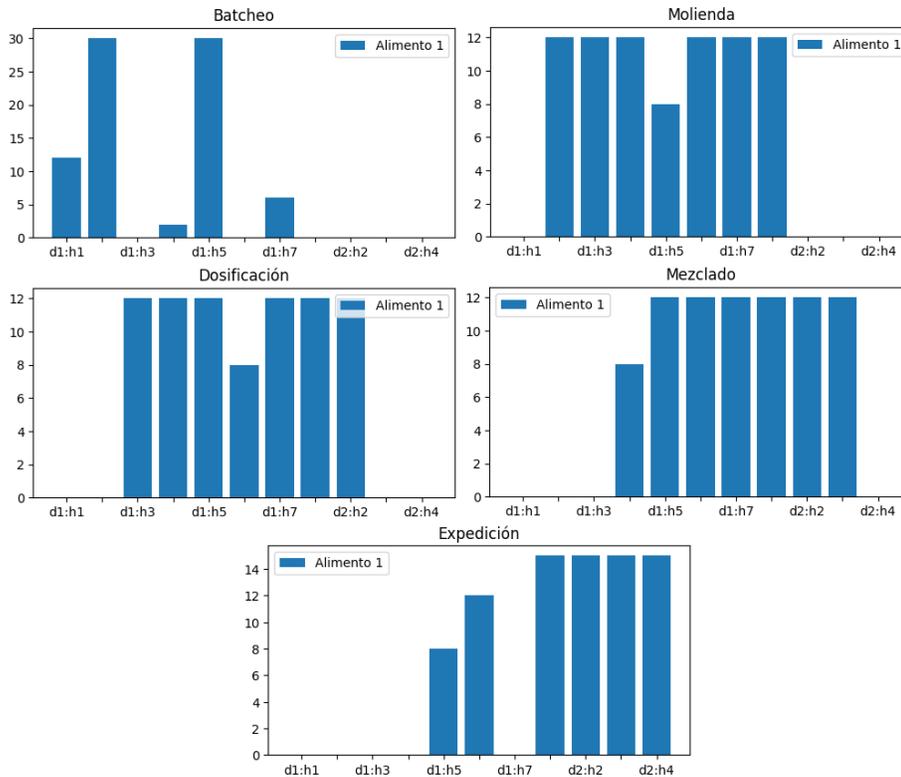


Figura 4.1: Resultados prueba 2 - primer modelado

4.1.4. Prueba 3

Descripción: Caso con demanda de más de un alimento.

Objetivo: Validar que el modelo funcione correctamente con más de un alimento, satisfaciendo la demanda y cumpliendo las restricciones de integridad para cada uno. Además, se busca que la cantidad total de alimento demandado sea menor que en la prueba anterior, de forma de que el modelo sea capaz de encontrar una solución óptima con tiempos de parada simultáneos en todas las etapas.

Demanda:

Id de alimento	Hora de entrega	Día de entrega	Cantidad
1	1	2	20
2	4	2	40

Tabla 4.4: Demanda prueba 3 - primer modelado.

Resultados: Solución óptima: 9 horas de actividad

A diferencia del caso anterior, en este caso el modelo fue capaz de encontrar una solución óptima que deja dos horas de inactividad en la planta. De las 11 horas disponibles para producir (entre el comienzo y la última hora de entrega), solamente necesitó de 9 horas de actividad para satisfacer la demanda. La figura 4.2 muestra la solución brindada por el modelo. Se puede observar como en las primeras dos horas del primer día ninguna de las etapas se encuentra activa.

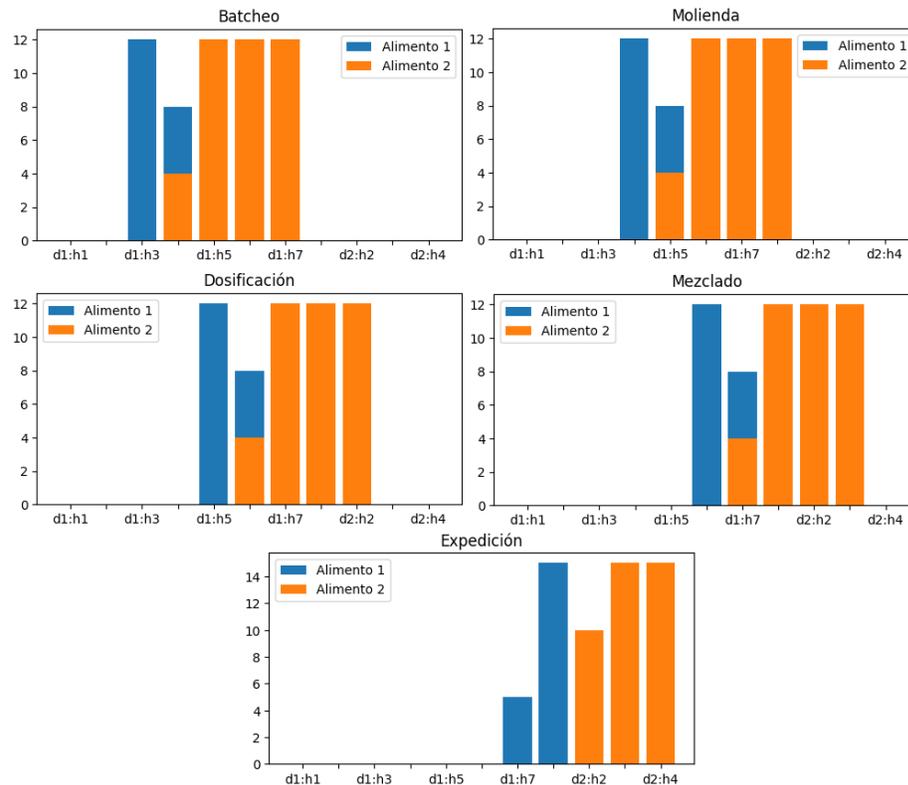


Figura 4.2: Resultados prueba 3 - primer modelado

4.2. Validación de correctitud del segundo modelado

4.2.1. Parámetros utilizados

- $D = 5$ (Cantidad de días disponibles para la producción)
- $H = 7$ (Cantidad de horas por día disponibles para la producción)
- Cap_i (Rendimiento de la etapa i en toneladas por hora)
 - $Cap_{etapa.0} = 12$
 - $Cap_{pellet} = 10$
 - $Cap_{exp.pell} = 14$
 - $Cap_{emb.pell} = 7$
 - $Cap_{exp.har} = 14$
 - $Cap_{emb.har} = 7$
- Emp_i (Cantidad de empleados que deben estar trabajando en la etapa i para que funcione)
 - $Emp_{etapa.0} = 4$
 - $Emp_{pellet} = 2$
 - $Emp_{exp} = 1$
 - $Emp_{emb} = 4$
- $CapAlm_i$ (Capacidad en toneladas de la tolva correspondiente a la etapa i)
 - $CapAlm_{exp.pell} = 30$
 - $CapAlm_{emb.pell} = 30$
 - $CapAlm_{exp.har} = 30$
 - $CapAlm_{emb.har} = 30$

4.2.2. Prueba 1

Descripción: Caso sencillo, con demanda de un solo alimento.

Objetivo: Probar el funcionamiento básico del segundo modelado con un caso sencillo.

Demanda:

Tipo	Hora entrega	Día entrega	Cantidad
Harina granel	4	2	30

Tabla 4.5: Demanda prueba 1 - segundo modelado.

Resultados: Solución óptima: 5 jornales requeridos.

Nuevamente, para mostrar las soluciones brindadas por el segundo modelado y el modelado final se utilizarán gráficos en lugar de una tabla con los resultados explícitos. En caso de querer consultar los resultados explícitos, los mismos se encuentran disponibles en el [anexo](#).

El primer gráfico muestra, para cada momento, la producción en la etapa 0 según el tipo del alimento. El segundo muestra la cantidad de alimento que se está produciendo en las etapas finales (expedición y embolsado). El tercer gráfico está dedicado al almacenamiento, mostrando la cantidad de alimento que se encuentra disponible en cada una de las tolvas en cada momento. Finalmente, el cuarto gráfico muestra las etapas que se encuentran activas en cada día.

La figura 4.3 resume la solución brindada por el modelo para la prueba 1. Como el objetivo del segundo modelado es minimizar la cantidad de jornales requeridos, este va a intentar que cada etapa se encuentre activa en la menor cantidad de días posibles. En este caso, como la demanda era de tipo “Harina granel”, las etapas “Etapa 0” y “Expedición harina” debían estar activas en algún momento. El modelo encontró una solución en la que cada una de las etapas se encuentra activa únicamente en el segundo día. El valor de la solución es de 5 jornales, ya que se requieren 4 para la etapa 0 y 1 para la etapa de expedición.

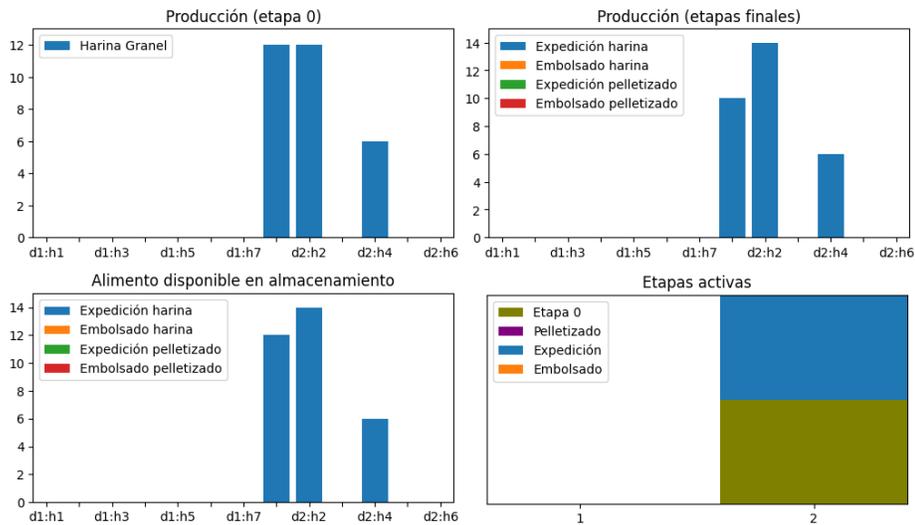


Figura 4.3: Resultados prueba 1 - segundo modelado

4.2.3. Prueba 2

Descripción: Demanda de dos alimentos con presentación de tipo harina pero distinto formato (granel y bolsas).

Objetivo: Comprobar que el modelo se comporta de la manera esperada ante demandas de distinto formato. Es esperable que, al ser alimentos de distinto formato, el modelo sea capaz de paralelizar la producción final (y el almacenamiento en las tolvas) de los mismos.

Demanda:

Tipo	Hora entrega	Día entrega	Cantidad
Harina granel	6	2	20
Harina bolsas	6	2	20

Tabla 4.6: Demanda prueba 2 - segundo modelado.

Resultados: Solución óptima: 9 jornales requeridos.

En este caso, nuevamente el modelo encontró una solución en la que cada etapa está activa solamente en un día. Se puede observar en la figura 4.4 como, al ser alimentos de distintos formato, fue posible paralelizar la producción y

almacenamiento de los mismos en las etapas finales. Esto debido a que la etapa final de ambos, y por consiguiente la tolva correspondiente, es distinta.

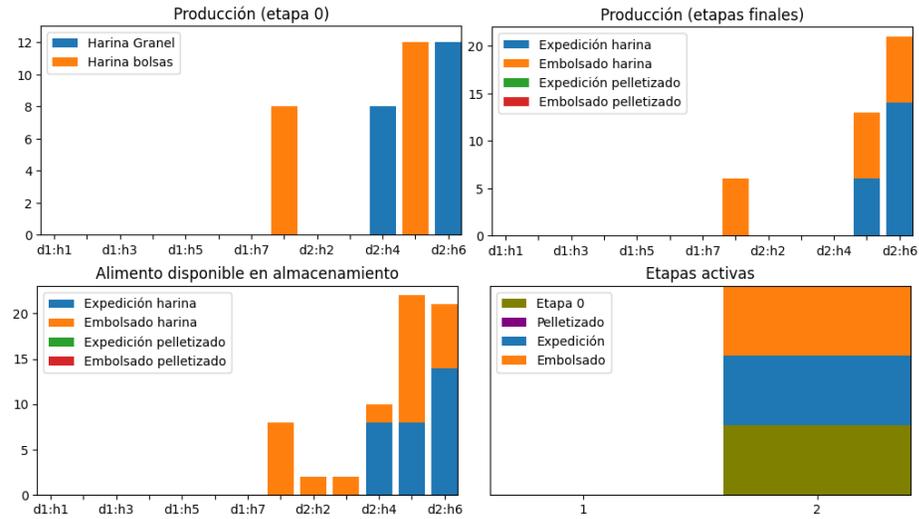


Figura 4.4: Resultados prueba 2 - segundo modelado

4.2.4. Prueba 3

Descripción: Demanda de dos alimentos con el mismo formato pero distinta presentación.

Objetivo: Similar a la prueba anterior, se busca validar que el modelo se comporta adecuadamente ante demandas con distinta presentación. En este caso, el modelo no debería ser capaz de paralelizar la producción ni el almacenamiento de los alimentos, ya que la etapa final de ambos es la misma.

Demanda:

Tipo	Hora entrega	Día entrega	Cantidad
Harina granel	4	2	20
Pelletizado granel	4	2	20

Tabla 4.7: Demanda prueba 3 - segundo modelado.

Resultados: Solución óptima: 7 jornales requeridos.

En este caso, otra vez el modelo encontró una solución en la que cada etapa está activa solamente en un día. En la figura 4.5 se puede notar como, a diferencia del caso anterior, no fue posible paralelizar la producción y almacenamiento de los alimentos en las etapas finales. Esto se da ya que, como los dos alientos tienen el mismo formato, la etapa final (y la tolva) es la misma para ambos (expedición). Por lo tanto, en cada momento solo se va a poder almacenar y producir uno de los dos alimentos.

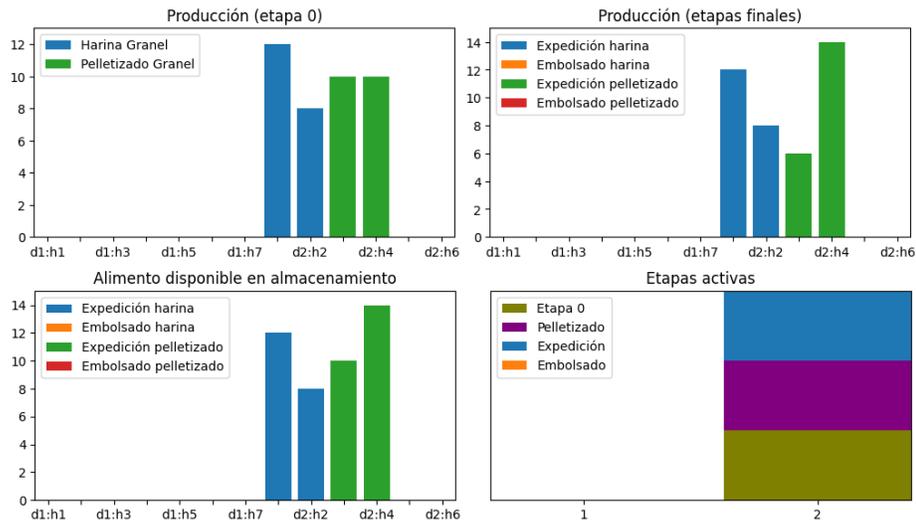


Figura 4.5: Resultados prueba 3 - segundo modelado

4.2.5. Prueba 4

Descripción: Demanda de dos alimentos en diferentes días.

Objetivo: Validar el funcionamiento del modelo ante demandas en distintos días. Se busca evaluar la capacidad del modelo de optimizar la cantidad de jornales requeridos. Más concretamente, evaluar si el modelo es capaz de utilizar las etapas que necesariamente se tienen que activar en el primer día (debido a la demanda en dicho día), para producir el alimento que debe entregarse en el segundo día, y así no activar dichas etapas en ambos días.

Demanda:

Tipo	Hora entrega	Día entrega	Cantidad
Pelletizado granel	6	1	20
Pelletizado bolsas	4	2	20

Tabla 4.8: Demanda prueba 4 - segundo modelado.

Resultados: Solución óptima: 11 jornales requeridos.

Al igual que en los casos anteriores, la solución brindada por el modelo no repite etapas entre días. Lo interesante de esta prueba es que, a diferencia de las pruebas anteriores, los alimentos deben ser entregados en diferentes días. Sin embargo, esto no implicó que la planta active todas las etapas en ambos días. Se puede observar en la figura 4.6 cómo, al haber capacidad disponible en el primer día en la etapa 0 y en la etapa de pelletizado, el modelo aprovecha esto para ya dejar el alimento almacenado en la tolva de embolsado (que es el alimento que debe ser entregado en el segundo día). De esta forma, en el segundo día solo debe activarse la etapa de embolsado, ya que el alimento ya había pasado por la etapa 0 y la etapa de pelletizado en el día anterior.

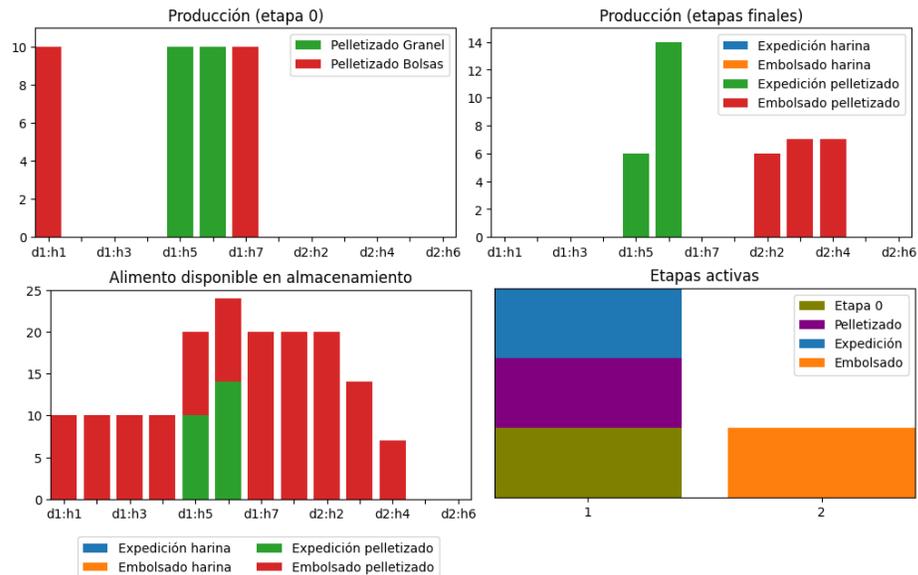


Figura 4.6: Resultados prueba 4 - segundo modelado

4.2.6. Prueba 5

Descripción: Prueba con demanda de todos los alimentos.

Objetivo: Evaluar el comportamiento del modelo ante una demanda compleja. Se busca someter al modelo a un caso más realista, con demandas de todos los tipos de alimentos, de forma de poder sacar conclusiones más generales del desempeño del mismo.

Demanda:

Tipo	Hora entrega	Día entrega	Cantidad
Harina granel	6	1	45
Harina bolsas	5	2	20
Pelletizado granel	2	2	30
Pelletizado bolsas	5	2	14

Tabla 4.9: Demanda prueba 5 - segundo modelado.

Resultados: Solución óptima: 17 jornales requeridos.

En esta instancia se pone a prueba el modelo bajo una demanda más exigente. Se puede observar en la figura 4.7 como el modelo fue capaz de mantener las etapas de expedición y embolsado activas en un único día. De esta forma, al procesar alimentos expedidos a granel en el primer día, y en bolsas en el segundo, logró que la cantidad de jornales requeridos no sea tan alta. Debido a que la demanda fue bastante alta, la etapa 0 y de pelletizado tuvieron que estar activas en ambos días.

Es interesante observar que en la solución brindada por el modelo, a la hora de separar la producción entre los días, lo hace según el formato del mismo (granel el primer día y bolsas el segundo). De esta forma, se obtiene un total de 17 jornales (8 por los dos días de la etapa 0, 4 por los dos días del pelletizado, 1 por la etapa de expedición y 4 por la etapa de embolsado). En caso de que se hubiera separado entre los días según la presentación del alimento (harinas y pelletizados), la cantidad de jornales requeridos hubiera sido mayor. Esto debido a que se hubiera requerido de 20 jornales (8 por los dos días de la etapa 0, 2 por el día de pelletizado, 8 por los dos días de embolsado y 2 por los dos días de expedición).

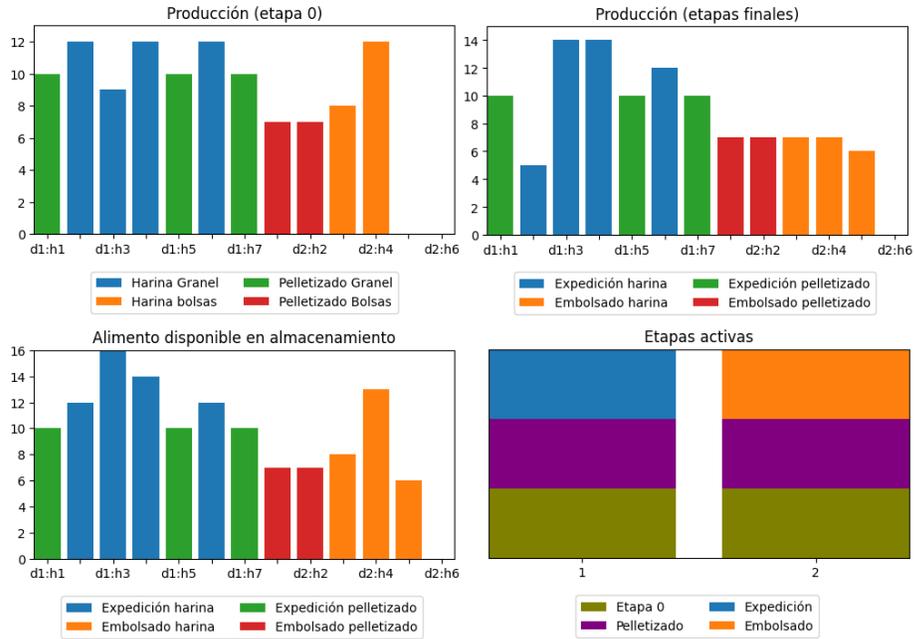


Figura 4.7: Resultados prueba 5 - segundo modelado

4.3. Validación de correctitud del modelado final

Como el segundo modelado y el modelado final comparten la misma estructura y funcionamiento de la planta, todas las pruebas realizadas para el segundo modelado aplican para este también. Por lo tanto, esta sección se centrará en evaluar la correctitud de las nuevas funcionalidades añadidas en el modelado final (tiempos de penalización e identificación de los alimentos), y luego una prueba del modelado ante una demanda semanal completa que refleje un caso de uso real. Cabe destacar que esta última demanda fue elaborada a partir de datos reales de la industria.

4.3.1. Parámetros utilizados

- $D = 5$ (Cantidad de días disponibles para la producción)
- $H = 7$ (Cantidad de horas por día disponibles para la producción)
- Cap_i (Rendimiento de la etapa i en toneladas por hora)

- $Cap_{etapa.0} = 12$
 - $Cap_{pellet} = 10$
 - $Cap_{exp.pell} = 14$
 - $Cap_{emb.pell} = 7$
 - $Cap_{exp.har} = 14$
 - $Cap_{emb.har} = 7$
- Emp_i (Cantidad de empleados que deben estar trabajando en la etapa i para que funcione)
- $Emp_{etapa.0} = 4$
 - $Emp_{pellet} = 2$
 - $Emp_{exp} = 1$
 - $Emp_{emb} = 4$
- $CapAlm_i$ (Capacidad en toneladas de la tolva correspondiente a la etapa i)
- $CapAlm_{exp.pell} = 30$
 - $CapAlm_{emb.pell} = 30$
 - $CapAlm_{exp.har} = 30$
 - $CapAlm_{emb.har} = 30$

4.3.2. Prueba 1

Descripción: Caso sencillo con dos alimentos.

Objetivo: Evaluar el modelo final ante una demanda de dos alimentos distintos. Concretamente, se busca validar que la penalización por cambio de alimento en la etapa 0 funciona correctamente.

Demanda:

Id	Tipo	Hora entrega	Día entrega	Cantidad
1	Harina granel	1	2	54
2	Harina bolsas	7	1	30

Tabla 4.10: Demanda prueba 1 - modelado final.

Resultados: Solución óptima: 14 jornales requeridos.

Para mostrar las soluciones de las pruebas del modelado final se utilizarán los mismos tipos de gráficos que en el segundo modelado. Como en este modelado, la granularidad del tiempo es de media hora, en los gráficos se mostrará una barra por cada media hora en lugar de una hora, como sucedía en los gráficos del modelado anterior. La otra diferencia es que en los gráficos de la producción y almacenamiento, para distinguir entre los distintos alimentos se usará el identificador del mismo, en lugar del tipo. Esto debido a que en este modelado pueden existir alimentos del mismo tipo (formato + presentación), pero con distinto identificador.

En la figura 4.8 se pueden observar los resultados de la prueba. Se puede comprobar cómo a la hora de cambiar el alimento producido en la etapa 0 (día 1, hora 3), hubo un tiempo de parada de media hora, satisfaciendo la restricción de penalización. Es interesante observar cómo, a raíz de esta penalización, no fue posible producir todo el alimento en el día 1.

Las cantidades de alimento demandadas fueron elegidas especialmente para que esto sucediera. Ya que al poder producir 6 toneladas cada media hora, la etapa 0 no es capaz de producir las 84 toneladas en 7 horas si existen tiempos de parada. De esta forma, la planta debió estar activa durante la primer media hora del día 2, lo que implicó un aumento de 5 jornales requeridos para satisfacer la demanda.

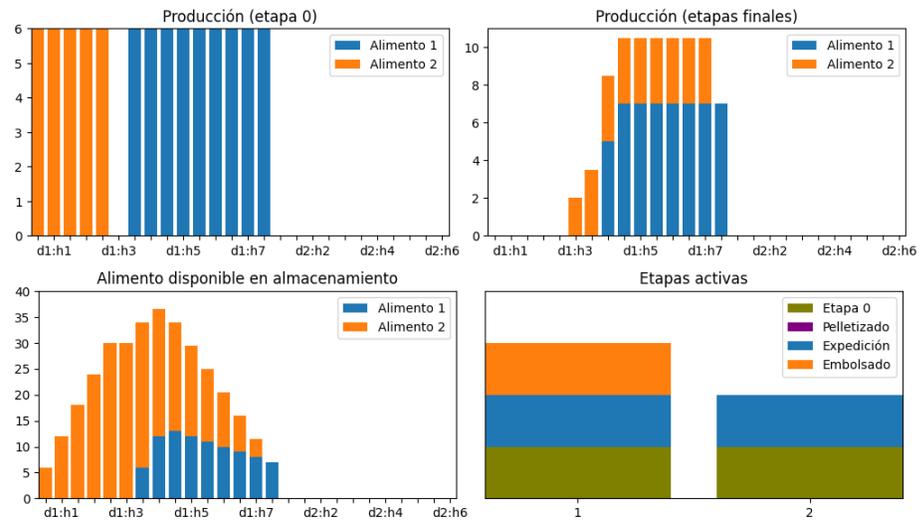


Figura 4.8: Resultados prueba 1 - modelado final

Datos de la ejecución:

Tiempo de ejecución	0,621 segundos
Número de variables	622

Tabla 4.11: Datos de ejecución prueba 1 - modelado final.

4.3.3. Prueba 2

Descripción: Demanda de alimentos del mismo tipo pero con distinto identificador.

Objetivo: Verificar que el modelo se comporta de manera adecuada ante demandas de distintos alimentos del mismo tipo. Se busca comprobar que en cada momento cada tolva almacene un único alimento, y que el modelo distinga entre los distintos identificadores a la hora de determinar la producción y el almacenamiento.

Demanda:

Id	Tipo	Hora entrega	Día entrega	Cantidad
1	Harina granel	6	1	30
2	Harina granel	7	1	30
3	Harina granel	5	2	12

Tabla 4.12: Demanda prueba 2 - modelado final.

Resultados: Solución óptima: 5 jornales requeridos.

La figura 4.9 ilustra los resultados de la prueba. Se puede observar cómo, al igual que en el caso anterior, se detuvo la producción en la etapa 0 durante media hora cada vez que cambió el alimento producido. Sin embargo, debido a la cantidad de alimento demandada, fue posible producir todo en el primer día. Además, se puede apreciar cómo el modelo respondió correctamente a las distintas demandas de los alimentos a pesar de ser todos del mismo tipo. Es decir, no mezcló los alimentos a la hora de producir o almacenar los mismos en las tolvas.

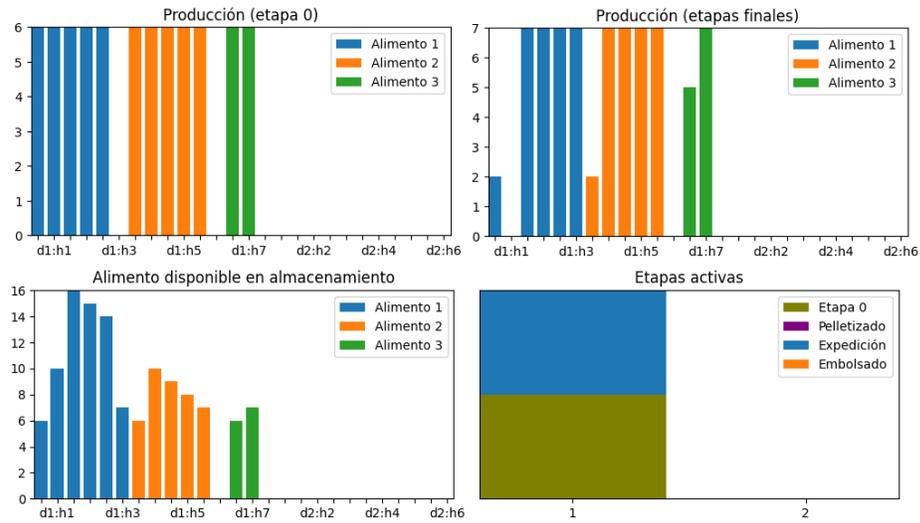


Figura 4.9: Resultados prueba 2 - modelado final

Datos de la ejecución:

Tiempo de ejecución	0,787 segundos
Número de variables	220

Tabla 4.13: Datos de ejecución prueba 2 - modelado final.

4.3.4. Prueba 3

Descripción: Caso realista con una demanda semanal completa. La demanda fue creada en base a los pedidos que tuvo una empresa real del rubro en los meses de Agosto y Septiembre del 2023, adaptando la misma a la capacidad de la planta modelada.

Objetivo: Probar el modelo final ante una demanda semanal completa, que refleje un caso de uso real.

Demanda:

Id	Tipo	Hora entrega	Día entrega	Cantidad
1	Harina granel	7	1	40
2	Harina bolsas	5	2	30
3	Pelletizado granel	6	2	40
4	Pelletizado bolsas	4	3	30
5	Harina granel	7	4	20
6	Pelletizado granel	6	5	30
7	Harina bolsas	6	5	30

Tabla 4.14: Demanda prueba 3 - modelado final.

Resultados: Solución óptima: 33 jornales requeridos.

En la figura 4.10 se resume la solución brindada por el modelo. A medida que la complejidad de las pruebas aumenta, agregando más demandas que se extienden en más días, las soluciones brindadas por el modelo se vuelven más ricas para el análisis. Al haber más alimentos demandados y más tiempo disponible para la producción, el modelo tiene más opciones a la hora de elegir el alimento que se produce en cada momento. A continuación se analizarán algunas de las decisiones tomadas por el modelo para llegar a esta solución óptima.

Es interesante observar cómo el modelo distribuyó la producción de alimentos en cada día. Por ejemplo, en el primer día la planta debía producir el alimento 1, ya que la demanda debía ser satisfecha a la hora 7 de dicho día. Por lo tanto debían estar activas la etapa 0 y la etapa de expedición necesariamente. El modelo aprovechó la capacidad disponible en la etapa 0 para dejar llena de alimento 2 la tolva correspondiente a la etapa de embolsado, cuya demanda debía ser satisfecha a la hora 5 del día siguiente.

Luego, en el segundo día, como las 30 toneladas que se debían producir de alimento 2 ya estaban en la tolva de embolsado, no fue necesario producir dicho alimento en la etapa 0. De esta forma se pudo aprovechar para producir el alimento 3 en la etapa 0, del cual se debían entregar 40 toneladas a la hora 6. Como el alimento 2 ya había pasado por la etapa 0 en el día anterior fue posible paralelizar la producción final de ambos alimentos, lo que permitió satisfacer ambas demandas. Finalmente, como las últimas horas del día se encontraban disponibles, se aprovechó para iniciar la producción del alimento 4, cuya demanda debía ser satisfecha en el día siguiente. Fue posible adelantar la producción del alimento 4 sin aumentar la cantidad de jornales requeridos porque las etapas requeridas para producirlo (etapa 0, pelletizado y embolsado) ya se habían activado en el día de todas formas.

En el día 3 el modelo terminó de producir el alimento 4, que debía entregarse

a la hora 3. Además, aprovechó que la etapa de embolsado estaba activa para ya dejar pronto todo el alimento 7, que debía entregarse en el día 5. De esta forma, todos los alimentos que requerían embolsado ya habían sido elaborados, y no se requeriría activar la etapa en los días siguientes.

Finalmente, el cuarto día se dedicó a la producción de los alimentos 5 y 6. Cuyas demandas debían ser satisfechas a la hora 7 del día 4 y a la hora 6 del día 5 respectivamente. De este modo fue posible no activar ninguna etapa de la planta en el día 5.

Datos de la ejecución:

Tiempo de ejecución	82,4 minutos
Número de variables	2165

Tabla 4.15: Datos de ejecución prueba 3 - modelado final.

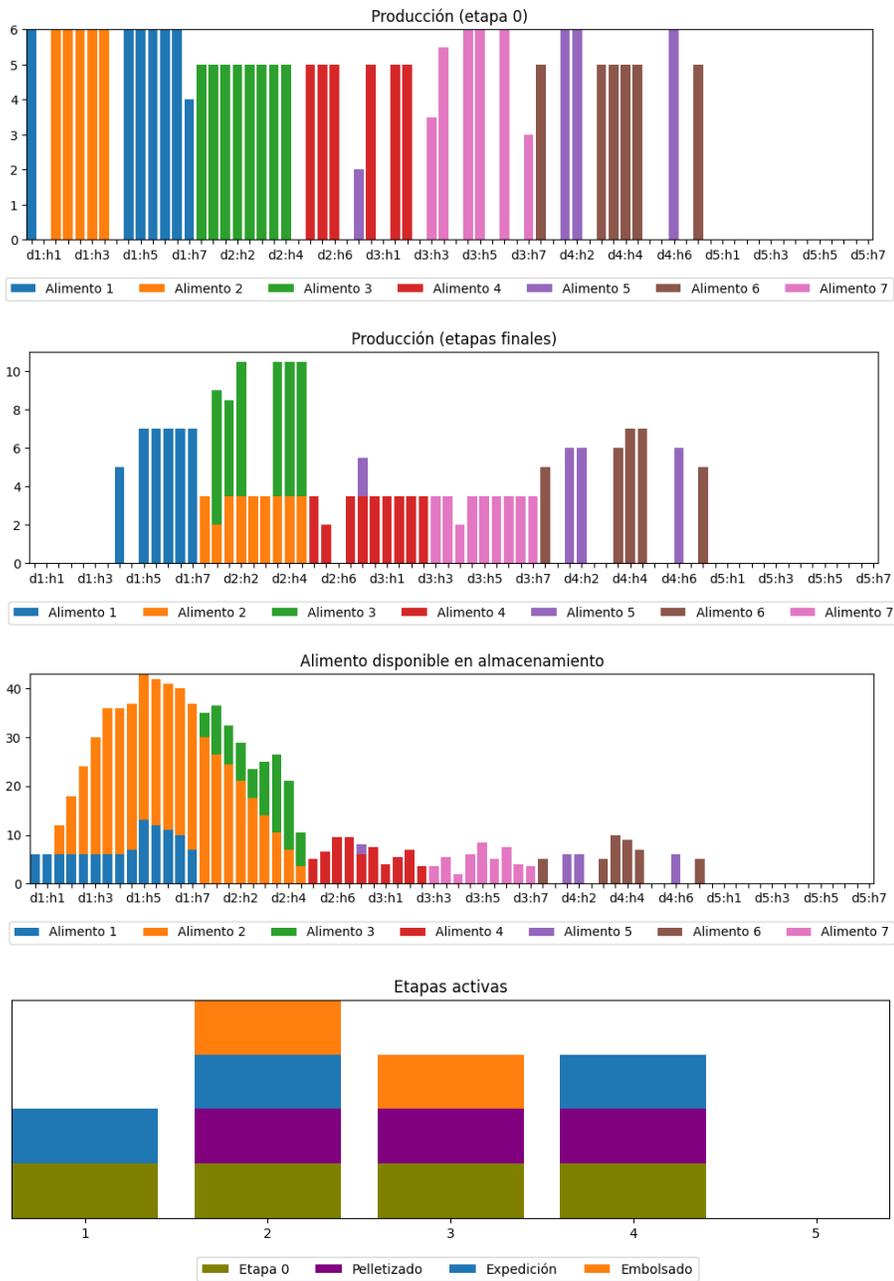


Figura 4.10: Resultados prueba 3 - modelado final

Capítulo 5

Pruebas de sensibilidad de parámetros

Para poder analizar el impacto que tiene cada uno de los parámetros sobre el modelo se realizaron pruebas de sensibilidad. Las mismas consisten en variar sensiblemente cada uno de los parámetros, para luego evaluar el impacto que tuvo dicha variación en los resultados brindados por el modelo. Las pruebas presentadas son únicamente para el modelado final, y los parámetros analizados son:

- El rendimiento de las etapas
- La capacidad de almacenamiento de las tolvas
- La cantidad de horas por día disponibles para la producción

Se considerarán las siguientes tres demandas para evaluar el rendimiento del modelo en cada prueba:

Demanda 1:

Id	Tipo	Hora entrega	Día entrega	Cantidad
1	Harina granel	7	1	30
2	Harina bolsas	5	2	25
3	Pelletizado granel	8	2	40
4	Pelletizado bolsas	7	3	40

Tabla 5.1: Demanda 1 - pruebas de sensibilidad de parámetros.

Demanda 2:

Id	Tipo	Hora entrega	Día entrega	Cantidad
1	Harina granel	5	1	20
2	Harina bolsas	8	1	20
3	Pelletizado granel	7	2	45
4	Pelletizado bolsas	4	3	45
5	Harina granel	6	4	40
6	Pelletizado bolsas	8	4	40

Tabla 5.2: Demanda 2 - pruebas de sensibilidad de parámetros.

Demanda 3:

Id	Tipo	Hora entrega	Día entrega	Cantidad
1	Harina granel	8	1	50
2	Harina bolsas	4	2	50
3	Pelletizado granel	7	2	30
4	Pelletizado bolsas	7	3	40
5	Harina bolsas	8	4	40
6	Pelletizado granel	7	5	40

Tabla 5.3: Demanda 3 - pruebas de sensibilidad de parámetros.

En cada análisis se presentarán los resultados de las pruebas. Mostrando, para cada demanda, la solución óptima (número de jornales requeridos) para cada opción del parámetro en cuestión. Además se especificará la variación del valor óptimo con respecto al parámetro usado previamente en las pruebas de correctitud. Cabe destacar que a la hora de evaluar el impacto de un determinado parámetro, el único parámetro que varía su valor es el que se está analizando, mientras que el resto mantienen el valor utilizado en las pruebas de correctitud.

5.1. Análisis del rendimiento de las etapas

Para analizar el impacto del rendimiento de las etapas se consideraron variaciones del parámetro utilizado en las pruebas de correctitud. Se tomaron en cuenta las siguientes opciones: rendimiento 15% por debajo del valor utilizado, el rendimiento utilizado en las pruebas de correctitud, y rendimiento 15% por encima del utilizado previamente.

En la tabla 5.4 se pueden ver los resultados de las pruebas. Como es de esperar, al aumentar el rendimiento de las etapas la cantidad de jornales requeridos para satisfacer al demanda descende. El incremento en el rendimiento de las etapas permite producir el alimento más rápido, lo que hace posible evitar la activación de etapas en ciertos días. Además, para la demanda 3 se puede observar cómo la baja en el rendimiento hace que el modelo no pueda encontrar una solución factible, es decir que la demanda no puede ser satisfecha de esa forma.

Demanda	Rendimiento	Sol. óptima	Variación
1	-15 %	26	+18,2 %
1	-	22	-
1	+15 %	21	-4,5 %
2	-15 %	43	+16,2 %
2	-	37	-
2	+15 %	32	-13,5 %
3	-15 %	∅ sol. factible	NA
3	-	41	-
3	+15 %	39	-4,9 %

Tabla 5.4: Resultados pruebas de sensibilidad del rendimiento de las etapas.

Sería interesante analizar en qué casos es rentable el incremento en el rendimiento de las etapas. Para esto sería necesario conocer el costo que conllevaría el incremento del rendimiento, para poder así comparar con la reducción de costos que implica la disminución de la cantidad de jornales. Análogamente se podría analizar en qué casos los ahorros que se podrían generar al disminuir el rendimiento de las etapas superan a los costos generados por el aumento en la cantidad de jornales requeridos.

Otro factor a considerar sería la cantidad de alimento que se demanda, ya que en casos de demandas altas no sería posible bajar el rendimiento de las etapas (al menos sin dejar demanda insatisfecha), mientras que el aumento del mismo permitiría satisfacer demandas mucho mayores.

5.2. Análisis de la capacidad de las tolvas

Para las pruebas del impacto de la capacidad de las tolvas se utilizaron los siguientes valores: 15 toneladas, 30 toneladas (valor utilizado para las pruebas de

correctitud) y 45 toneladas. Para simplificar, en cada prueba se utilizó el mismo valor de capacidad para ambas tolvas (expedición y embolsado), pero también hubiera sido posible experimentar variando de forma distinta la capacidad de cada tolva.

Antes de analizar los resultados de las pruebas es interesante reflexionar sobre la importancia de las tolvas y su capacidad en el modelo. Las tolvas cumplen la función de almacenar el alimento previo a las etapas de expedición y embolsado, por lo que permiten procesar el alimento en las etapas anteriores y dejarlo almacenado para terminar de procesarlo después. A los efectos del modelo, podrían no existir las tolvas, y que el alimento deba producirse inmediatamente una vez está disponible, aunque la planta sería bastante menos eficiente. Dicho esto, a diferencia de lo que sucede con los otros parámetros analizados en este capítulo, aumentar la capacidad de las tolvas no siempre va a verse reflejado en un aumento del rendimiento de la planta. Esto debido a que, llegado a cierto punto de capacidad, el seguir aumentándola no va a tener ningún impacto, porque de todas formas el modelo no va a ser capaz de utilizar esta capacidad de almacenamiento extra. Un ejemplo claro sería si se tienen tolvas con más capacidad que todas las demandas de alimento: nunca se va a utilizar esa capacidad extra porque nunca va a haber necesidad de producir esas cantidades de alimento.

Los resultados de las pruebas se muestran en la tabla 5.5. Se puede ver cómo lo mencionado en el párrafo anterior aplica directamente. Para las demandas 1 y 2, el reducir la capacidad de las tolvas a 15 toneladas implica una degradación en el rendimiento de la planta; mientras que aumentarla a 45 no tiene ningún impacto. Mientras tanto, para la demanda 3 se obtiene exactamente el mismo resultado con 15, 30 o 45 toneladas de capacidad.

Demanda	Capacidad	Sol. óptima	Variación
1	15	25	+13,6 %
1	30	22	-
1	45	22	0 %
2	15	41	+10,8 %
2	30	37	-
2	45	37	0 %
3	15	41	0 %
3	30	41	-
3	45	41	0 %

Tabla 5.5: Resultados pruebas de sensibilidad de la capacidad de las tolvas.

5.3. Análisis de la cantidad de horas

Para el análisis sobre el impacto de la cantidad de horas por día disponibles para la producción en los resultados del modelo, se consideraron tres opciones: 6 horas, 7 horas (parámetro elegido para las pruebas de correctitud) y 8 horas.

En la tabla 5.6 se encuentran los resultados de las pruebas. Se puede observar cómo, para las tres demandas, al aumentar la cantidad de horas que trabajan los empleados, la cantidad de jornales requeridos disminuye. Además, en el caso de la demanda 3, si los empleados trabajan solamente 6 horas por día, no es posible satisfacer la demanda.

Además de incluir los valores óptimos para cada caso, se añadió la cantidad total de horas que fueron requeridas. Es decir, la cantidad de jornales requeridos multiplicado por la cantidad de horas de cada jornal. De esta forma es posible comprobar si la disminución en la cantidad de jornales se vio traducida en una disminución también de la cantidad total de horas de trabajo requeridas. En este caso se puede observar cómo los resultados no son consistentes, ya que para algunas configuraciones la cantidad total de horas disminuye al aumentar el largo del jornal, mientras que para otras aumenta.

Demanda	Horas	Sol. óptima	Var. sol.	# total de horas	Var. # horas
1	6	26	+18,2 %	156	+1,3 %
1	7	22	-	154	-
1	8	21	-4,5 %	168	+9,1 %
2	6	43	+16,2 %	258	-0,4 %
2	7	37	-	259	-
2	8	32	-13,5 %	256	-1,2 %
3	6	∄ sol. factible	NA	NA	NA
3	7	41	-	287	-
3	8	39	-4,9 %	312	+8,7 %

Tabla 5.6: Resultados pruebas de sensibilidad del parámetro horas.

Se podría analizar en qué casos la disminución en la cantidad de jornales requeridos compensa el aumento en la duración del jornal. En los casos en que la cantidad total de horas de trabajo son menores al aumentar el número de horas trabajadas por jornal, se podría suponer que la decisión de aumentar el número de horas tendría sentido, aunque para esto habría que considerar el costo de la hora de trabajo en cada una de las opciones, lo cual escapa al alcance de

este trabajo.

Sin dudas otro factor a considerar sería la cantidad de alimento que se demanda. De forma similar a lo que ocurría con el rendimiento de las etapas, en casos de demandas altas la reducción de las horas del jornal podría no ser posible (sin dejar demanda insatisfecha), mientras que el aumento en el largo del jornal daría lugar a una capacidad de producción notablemente mayor.

Capítulo 6

Consideraciones finales

6.1. Conclusiones

A lo largo del proyecto se logró definir un problema de optimización orientado a la planificación semanal de la producción de alimentos balanceados. Se obtuvo un modelo matemático de programación lineal para la resolución del problema, el cual se probó con distintas instancias del mismo.

Se considera que el modelo implementado refleja la realidad de forma satisfactoria. Si bien el modelo, como toda abstracción matemática de la realidad, difícilmente logre ser un reflejo exacto de la misma, se cree que el mismo alcanza a captar un nivel de detalle de acuerdo a las expectativas planteadas en un principio.

Además, por cómo fue implementado el modelo, el mismo es altamente adaptable a otro tipo de plantas de producción de alimentos. Los valores para el rendimiento de las etapas, cantidad de horas y días disponibles para la producción, capacidad de las tolvas, y cantidad de empleados que deben trabajar en cada una de las etapas son todos parámetros fácilmente configurables. A la vez, si fuera necesario agregar una etapa nueva, o quitar alguna de las existentes, se podría hacer sencillamente agregando o modificando alguna de las restricciones. Esto aplica también al caso en que una planta disponga de más de una estación para alguna de las etapas que el modelo ya considera. Por ejemplo, si una planta tiene dos estaciones de pelletizado, que permiten el procesamiento en paralelo de distintos tipos de alimentos, no sería difícil adaptar el modelo para captar este cambio.

El modelo fue probado extensivamente mediante numerosas pruebas, algunas de las cuales fueron incluidas en este documento. Mediante las pruebas de correctitud se logró validar el correcto funcionamiento del modelo; mientras que con las pruebas de sensibilidad de parámetros se analizó el impacto de cada uno de ellos en el modelo, logrando entender mejor qué efecto podrían tener las variaciones en cada uno de estos parámetros.

Si bien no se realizó un análisis de la performance del modelo (es uno de los trabajos futuros planteados), el mismo respondió de forma correcta en todas las pruebas realizadas. En la mayoría de las pruebas, el modelo fue capaz de encontrar una solución óptima en escasos minutos, mientras que para los casos más complejos, se extendió sensiblemente, pero sin llegar a superar los 90 minutos en ningún caso.

6.2. Desvíos con respecto a la propuesta inicial

El cambio principal con respecto a la propuesta original fue sobre la función objetivo a optimizar. Inicialmente se había propuesto “minimizar el tiempo productivo, y por ende los costos energéticos y de mano de obra”, pero en cierto punto, junto a los tutores se decidió que podía ser más interesante minimizar la cantidad de jornales requeridos. Si se presta atención al primer modelado, se puede observar que en él la función a minimizar era la cantidad de horas de actividad de la planta; mientras que a partir del segundo modelado se busca minimizar la cantidad de jornales. De todas formas, el cambio necesario para pasar a minimizar la cantidad de horas de actividad en el segundo modelado y el modelado final sería un sencillo ajuste, simplemente pareció más interesante el análisis minimizando la cantidad de jornales.

También hubo ciertos desvíos en relación al [cronograma de trabajo](#) planteado en un principio. En la tabla [6.1](#) se pueden observar las tareas del cronograma, junto a la duración que se había planificado y el tiempo que efectivamente se les dedicó.

Tarea	Duración estimada	Tiempo dedicado
Relevamiento bibliográfico y armado del estado del arte con problemas relacionados.	2 meses	2 meses
Modelización e Implementación del Problema.	1.5 meses	3 meses
Testeo del Modelo desarrollado usando herramientas de optimización como ser CPLEX.	1.5 meses	2.5 meses
Estudio de sensibilidad de la solución frente a variaciones en los parámetros. Analizar distintas combinaciones de parámetros y sus resultados	1 mes	1 mes
Documentación del proyecto.	2 meses	3 meses

Tabla 6.1: Reporte de trabajo realizado

El desvío total fue de unos 3 meses y medio, y analizándolo en retrospectiva se pueden encontrar algunas de sus causas. Una primer idea es que se subestimó levemente el tiempo estimado para el modelado e implementación del sistema. Seguramente un estimado de 2 meses hubiera sido más acertado que el mes y medio que se estimó. De todas formas, se entiende que el principal causante del desvío fue el no poder dedicar la cantidad de horas pautadas al proyecto en determinados momentos. Por motivos personales y de superposición con otras actividades de la carrera, durante algunas semanas no fue posible dedicarle al proyecto las 15 horas semanales que se espera. Justamente, dichas semanas coinciden con las correspondientes a tareas en las cuales hubo desvíos con respecto al tiempo estimado

6.3. Trabajo futuro

Con base en los resultados obtenidos y las conclusiones derivadas de este proyecto, se pueden identificar varias oportunidades y áreas para la continuación y expansión del trabajo realizado. A continuación, se presentan las algunas posibles líneas de trabajo que pueden ser exploradas en el futuro:

- **Análisis cruzado de sensibilidad de parámetros:** En el análisis de sensibilidad de parámetros se realizaron pruebas variando un parámetro a la vez. Sería interesante hacer un análisis sobre qué sucede al variar más de un parámetro al mismo tiempo. Por ejemplo, disminuir la cantidad de horas del jornal, pero aumentando el rendimiento de las etapas. De esta forma se podría estudiar qué tanto se debería incrementar el rendimiento

de las etapas para poder reducir el largo del jornal sin afectar la productividad de la planta.

- **Pruebas de performance del modelo:** Las pruebas realizadas al modelo estuvieron centradas en el desempeño del mismo, entendido como los resultados que brinda ante distintas demandas, y bajo distintas configuraciones. Una idea para un trabajo futuro sería agregar pruebas de performance, que evalúen los tiempos de ejecución y recursos utilizados por el modelo ante distintos casos. Estas pruebas fueron relegadas en este proyecto debido a que los casos analizados pudieron ser resueltos con escasos recursos, y en no mucho tiempo (minutos en la mayoría de los casos). De todas formas, el análisis de performance puede ser de interés para casos de plantas con una mayor infraestructura, que acepten demandas mucho más complejas. También podría ser útil en casos en los que el tiempo de respuesta es un factor crítico.
- **Empleados no especializados:** Durante el modelado de la realidad, se asumió que los empleados de la planta iban a ser especializados. Es decir que si se los contrata para trabajar en una determinada etapa, solo van a poder trabajar en esa etapa. Una alternativa sería que los empleados no fueran especializados, de esta forma podrían trabajar en más de una etapa en un día (aunque no simultáneamente). Esto implicaría cambios en las restricciones, ya que la cantidad de jornales requeridos ya no quedaría determinada por las etapas que se activan en cada día, sino por las etapas que se activan simultáneamente en cada día.
- **Optimizar ganancias:** Una idea interesante que surgió en el transcurso del proyecto, pero que no se llegó a implementar debido a la complejidad que implica, es la de modificar la función objetivo para que esta sea directamente maximizar las ganancias de la planta. De esta manera, lo que se buscaría sería minimizar los costos de la planta, ya que los ingresos estarían determinados de antemano por la demanda de alimentos. Claramente el número de jornales requeridos tiene un impacto en los costos, pero no es el único factor. Por ejemplo, la cantidad de horas de actividad también influye en el costo. Además, el pasar a optimizar las ganancias permite incluir muchos otros aspectos en la función objetivo. A modo de ejemplo, se podría considerar la opción de contratar empleados especializados y no especializados, cada uno a un costo distinto. También se podría considerar la opción de que el rendimiento de las etapas sea variable de acuerdo al costo de producción por hora, ya sea porque la cantidad de empleados que se le puede asignar es variable, o porque puede funcionar a partir de distintos insumos. Así, el modelo buscaría la combinación de rendimiento/costo que mejor funciona para cada caso.
- **Contemplar demandas insatisfechas:** De la mano del punto anterior, también se podría considerar la opción de contemplar demandas insatisfechas. Pueden existir casos en que a la planta no le convenga cumplir

con cierta demanda, porque por ejemplo esto podría implicar que la planta tenga que activarse en un día solamente para producir una cantidad pequeña de alimento. La idea de este punto es que, a diferencia del punto anterior, los ingresos de la planta también sean variables, en lugar de quedar determinado por la demanda. Una opción sería permitir que existan demandas insatisfechas cuando al modelo no le convenga hacerlo; otra opción sería permitir que el modelo las satisfaga en otro momento, por ejemplo postergando demandas pero disminuyendo los ingresos percibidos por las mismas.

Referencias

- Castillo, D. M., y Carvajal, J. (2019). Modelo de asignación de recursos aplicado al centro de diseño tecnológico industrial del sena en cali. *Universidad autónoma de Occidente*.
- Chachapoya, D. L. (2014). Producción de alimentos balanceados en una planta procesadora en el cantón cevallos. *Escuela Politécnica Nacional*.
- Galindo, M. (s.f.). Producción de alimentos apoyada con programación lineal. *Universidad Rafael Landívar*.
- IBM. (2021). *Setting up the Python API of CPLEX*. <https://www.ibm.com/docs/en/icos/20.1.0?topic=cplex-setting-up-python-api>. (Accessed: 2023-12-06)
- IBM. (2023). *IBM ILOG CPLEX optimization studio*. <https://www.ibm.com/es-es/products/ilog-cplex-optimization-studio>. (Accessed: 2023-11-19)
- IBM Decision Optimization CPLEX Modeling for Python [Manual de software informático]. (2022). <http://ibmdecisionoptimization.github.io/docplex-doc/>.
- Jupyter. (2023). *Jupyter Notebook Documentation*. <https://jupyter-notebook.readthedocs.io/en/latest/>.
- Rivas, C., y Leonardo, D. (2013). Producción de alimentos balanceados en una planta procesadora en el cantón Cevallos. *Escuela Politécnica Nacional*.
- Valencia, V. R., Aguirre, D. C., y Herrera, S. R. (2018). Programación o planeación de actividades o recursos en la agricultura. Una revisión de literatura. *Universidad Nacional de Colombia*.
- Vanderbei, R. J. (2020). *Linear programming*. Springer Cham. doi: 10.1007/978-3-030-39415-8
- Vargas, R., y Gutierrez, J. E. (2018). Aplicación de la programación lineal para optimizar el costo de una dieta balanceada. *INGnosis*.
- Williams, H. P. (2009). *Integer programming*. Springer US. doi: 10.1007/978-0-387-92280-5.2

Anexo A

Código Python del modelo final

```
1 from enum import Enum
2 from itertools import product
3
4 import matplotlib.pyplot as plt
5 import pandas as pd
6 from docplex.mp.model import Model
7
8
9 #####
10 # Constantes
11 #####
12 GRANULARIDAD_HORA = 2
13 CANTIDAD_HORAS = 7
14 CANTIDAD_DIAS = 5
15
16 MULT_RENDIMIENTO_ETAPA = 1
17 CAPACIDAD_TOLVAS = 30
18
19 # Archivo CSV con las siguientes columnas:
20 # id,tipo_alimento,hora,dia,cantidad
21 DEMANDAS_FILE = "demandas.csv"
22
23 # Tiempo limite del modelo (en segundo), si definido
24 MODEL_TIME_LIMIT = None
25
26
27 #####
28 # Etapas
29 #####
30 class Etapa(str, Enum):
31     etapa_0 = "Etapa 0"
```

```

32     pelletizado = "Pelletizado"
33     expedicion_pelletizado = "Expedición pelletizado"
34     embolsado_pelletizado = "Embolsado pelletizado"
35     expedicion_harina = "Expedición harina"
36     embolsado_harina = "Embolsado harina"
37
38
39     class EtapaReal(str, Enum):
40         etapa_0 = "Etapa 0"
41         pelletizado = "Pelletizado"
42         expedicion = "Expedición"
43         embolsado = "Embolsado"
44
45
46     #####
47     # Días y horas
48     #####
49     horas = [(hora + 1) for hora in range(CANTIDAD_HORAS * GRANULARIDAD_HORA)]
50     dias = [(dia + 1) for dia in range(CANTIDAD_DIAS)]
51
52
53     #####
54     # Capacidad de etapas y tolvas
55     #####
56     capacidad_etapas = {
57         EtapaReal.etapa_0: 12,
58         EtapaReal.pelletizado: 10,
59         EtapaReal.expedicion_pelletizado: 14,
60         EtapaReal.embolsado_pelletizado: 7,
61         EtapaReal.expedicion_harina: 14,
62         EtapaReal.embolsado_harina: 7,
63     }
64     capacidad_etapas = {
65         etapa: capacidad * MULT_RENDIMIENTO_ETAPA
66         for etapa, capacidad in capacidad_etapas.items()
67     }
68
69     empleados_etapa = {
70         EtapaReal.etapa_0: 4,
71         EtapaReal.pelletizado: 2,
72         EtapaReal.expedicion: 1,
73         EtapaReal.embolsado: 4,
74     }
75
76     capacidad_almacenamiento = {
77         EtapaReal.expedicion_harina: CAPACIDAD_TOLVAS,
78         EtapaReal.embolsado_harina: CAPACIDAD_TOLVAS,
79         EtapaReal.expedicion_pelletizado: CAPACIDAD_TOLVAS,
80         EtapaReal.embolsado_pelletizado: CAPACIDAD_TOLVAS,
81     }
82
83
84     #####
85     # Demanda y tipos de alimentos
86     #####
87     class TipoAlimento(str, Enum):
88         harina_granel = "Harina Granel"

```

```

89     harina_bolsas = "Harina Bolsas"
90     pelletizado_granel = "Pelletizado Granel"
91     pelletizado_bolsas = "Pelletizado Bolsas"
92
93
94     # Para cada tipo de alimento, la última etapa correspondiente
95     tipo_alimento_etapa = {
96         TipoAlimento.harina_granel: Etapa.expedicion_harina,
97         TipoAlimento.harina_bolsas: Etapa.embolsado_harina,
98         TipoAlimento.pelletizado_granel: Etapa.expedicion_pelletizado,
99         TipoAlimento.pelletizado_bolsas: Etapa.embolsado_pelletizado,
100    }
101
102    demandas_df = pd.read_csv(DEMANDAS_FILE)
103    demanda = {}
104    for _, row in demandas_df.iterrows():
105        demanda[row.id] = (
106            TipoAlimento(row.tipo_alimento),
107            (row.hora, row.dia),
108            row.cantidad,
109        )
110    alimentos = demanda.keys()
111
112
113    #####
114    # Modelo y variables
115    #####
116    model = Model(name="alimentos_v3.0")
117
118    prod_range = [
119        (a, i, h, d) for a in alimentos for i in Etapa for h in horas for d in dias
120    ]
121    alm_range = [
122        (a, i, h, d)
123        for a in alimentos
124        for i in capacidad_almacenamiento.keys()
125        for h in horas
126        for d in dias
127    ]
128    empl_act_range = [(i, d) for i in EtapaReal for d in dias]
129    prod_bool_range = [
130        (a, i, h, d) for a in alimentos for i in Etapa for h in horas for d in dias
131    ]
132    alm_libre_range = [
133        (a, i, h, d)
134        for a in alimentos
135        for i in capacidad_almacenamiento.keys()
136        for h in horas
137        for d in dias
138    ]
139
140    produccion = model.continuous_var_dict(keys=prod_range, name="produccion")
141    almacenamiento = model.continuous_var_dict(
142        keys=alm_range, name="almacenamiento"
143    )
144    empleados_activos = model.integer_var_dict(keys=empl_act_range, name="activa")
145    prod_bool = model.binary_var_dict(keys=prod_bool_range, name="prod_bool")

```

```

146 alm_libre = model.binary_var_dict(keys=alm_libre_range, name="alm_libre")
147
148
149 #####
150 # Función objetivo
151 #####
152 model.minimize(model.sum(empleados_activos))
153 if MODEL_TIME_LIMIT:
154     model.set_time_limit(MODEL_TIME_LIMIT)
155
156
157 #####
158 # Restricciones
159 #####
160 #####
161 # Restricciones inter-variables
162 #####
163 # Indicador para `alm_libre`
164 for alimento in alimentos:
165     for etapa in capacidad_almacenamiento.keys():
166         for hora in horas:
167             for dia in dias:
168                 horas_previas = [h for h in horas if h < hora]
169                 horas_previas_inc = [h for h in horas if h <= hora]
170                 dias_previos = [d for d in dias if d < dia]
171                 model.add_indicator(
172                     alm_libre[(alimento, etapa, hora, dia)],
173                     model.sum(
174                         almacenamiento[(alimento, etapa, h, d)]
175                         for h in horas
176                         for d in dias_previos
177                     )
178                     + model.sum(
179                         almacenamiento[(alimento, etapa, h, dia)]
180                         for h in horas_previas_inc
181                     )
182                     - model.sum(
183                         produccion[(alimento, etapa, h, d)]
184                         for h in horas
185                         for d in dias_previos
186                     )
187                     - model.sum(
188                         produccion[(alimento, etapa, h, dia)]
189                         for h in horas_previas
190                     )
191                     == 0,
192                 )
193
194 # Empleados activos
195 for i in [Etapa.etapa_0, Etapa.pelletizado]:
196     for d in dias:
197         model.add_if_then(
198             model.sum(prod_bool[(a, i, h, d)] for a in alimentos for h in horas)
199             >= 1,
200             empleados_activos[(i, d)] == empleados_etapa[i],
201         )
202 for d in dias:

```

```

203     model.add_if_then(
204         model.sum(
205             prod_bool[(a, Etapa.expedicion_harina, h, d)]
206             for a in alimentos
207             for h in horas
208         )
209         + model.sum(
210             prod_bool[(a, Etapa.expedicion_pelletizado, h, d)]
211             for a in alimentos
212             for h in horas
213         )
214         >= 1,
215         empleados_activos[(EtapaReal.expedicion, d)]
216         == empleados_etapa[EtapaReal.expedicion],
217     )
218 for d in dias:
219     model.add_if_then(
220         model.sum(
221             prod_bool[(a, Etapa.embolsado_harina, h, d)]
222             for a in alimentos
223             for h in horas
224         )
225         + model.sum(
226             prod_bool[(a, Etapa.embolsado_pelletizado, h, d)]
227             for a in alimentos
228             for h in horas
229         )
230         >= 1,
231         empleados_activos[(EtapaReal.embolsado, d)]
232         == empleados_etapa[EtapaReal.embolsado],
233     )
234
235 #####
236 # Atajo
237 #####
238 def get_alimento_disponible(alimento, etapa, hora, dia):
239     horas_previas = [h for h in horas if h < hora]
240     horas_previas_inc = [h for h in horas if h <= hora]
241     dias_previos = [d for d in dias if d < dia]
242     return (
243         model.sum(
244             almacenamiento[(alimento, etapa, h, d)]
245             for h in horas
246             for d in dias_previos
247         )
248         + model.sum(
249             almacenamiento[(alimento, etapa, h, dia)] for h in horas_previas_inc
250         )
251         - model.sum(
252             produccion[(alimento, etapa, h, d)]
253             for h in horas
254             for d in dias_previos
255         )
256         - model.sum(
257             produccion[(alimento, etapa, h, dia)] for h in horas_previas
258         )
259     )

```

```

260
261 #####
262 # Restricciones del problema
263 #####
264 # Demanda satisfecha
265 for alimento, (
266     tipo_alimento,
267     (hora, dia),
268     cantidad,
269 ) in demanda.items():
270     horas_previas_inc = [h for h in horas if h <= hora * GRANULARIDAD_HORA]
271     dias_previos = [d for d in dias if d < dia]
272     model.add_constraint(
273         model.sum(
274             produccion[
275                 (
276                     alimento,
277                     tipo_alimento_etapa[tipo_alimento],
278                     h,
279                     d,
280                 )
281             ]
282             for h in horas
283             for d in dias_previos
284         )
285         + model.sum(
286             produccion[
287                 (
288                     alimento,
289                     tipo_alimento_etapa[tipo_alimento],
290                     h,
291                     dia,
292                 )
293             ]
294             for h in horas_previas_inc
295         )
296         == cantidad
297     )
298
299 # Etapas no desbordadas
300 for a in alimentos:
301     for i in Etapa:
302         for h in horas:
303             for d in dias:
304                 model.add_constraint(
305                     produccion[(a, i, h, d)]
306                     <= (
307                         prod_bool[(a, i, h, d)]
308                         * (capacidad_etapas[i] / GRANULARIDAD_HORA)
309                     )
310                 )
311
312 # Almacenamiento no desbordado
313 for a in alimentos:
314     for i in tipo_alimento_etapa.values():
315         for h in horas:
316             for d in dias:

```

```

317         model.add_constraint(
318             get_alimento_disponible(a, i, h, d)
319             <= capacidad_almacenamiento[i]
320         )
321
322     # Produccion no desbordada
323     for a in alimentos:
324         for i in tipo_alimento_etapa.values():
325             for h in horas:
326                 for d in dias:
327                     model.add_constraint(
328                         produccion[(a, i, h, d)]
329                         <= get_alimento_disponible(a, i, h, d)
330                     )
331
332     # Flujo
333     for a in alimentos:
334         for h in horas:
335             for d in dias:
336                 model.add_constraint(
337                     sum(
338                         [
339                             produccion[(a, Etapa.pelletizado, h, d)],
340                             almacenamiento[(a, Etapa.expedicion_harina, h, d)],
341                             almacenamiento[(a, Etapa.embolsado_harina, h, d)],
342                         ]
343                     )
344                     == produccion[(a, Etapa.etapa_0, h, d)]
345                 )
346
347     # Flujo pelletizado
348     for a in alimentos:
349         for h in horas:
350             for d in dias:
351                 model.add_constraint(
352                     almacenamiento[(a, Etapa.expedicion_pelletizado, h, d)]
353                     + almacenamiento[(a, Etapa.embolsado_pelletizado, h, d)]
354                     == produccion[(a, Etapa.pelletizado, h, d)]
355                 )
356
357     # Mutex harina-pelletizado
358     for a1 in alimentos:
359         for a2 in alimentos:
360             for h in horas:
361                 for d in dias:
362                     if a1 != a2:
363                         model.add_if_then(
364                             alm_libre[a1, Etapa.expedicion_harina, h, d] == 0,
365                             almacenamiento[(a2, Etapa.expedicion_harina, h, d)]
366                             == 0,
367                         )
368                         model.add_if_then(
369                             alm_libre[a1, Etapa.expedicion_harina, h, d] == 0,
370                             almacenamiento[
371                                 (
372                                     a2,
373                                     Etapa.expedicion_pelletizado,

```

```

374         h,
375         d,
376     )
377 ]
378 == 0,
379 )
380 model.add_if_then(
381     alm_libre[
382         (
383             a2,
384             Etapa.expedicion_pelletizado,
385             h,
386             d,
387         )
388     ]
389     == 0,
390     almacenamiento[
391         (
392             a1,
393             Etapa.expedicion_pelletizado,
394             h,
395             d,
396         )
397     ]
398     == 0,
399 )
400 model.add_if_then(
401     alm_libre[
402         (
403             a2,
404             Etapa.expedicion_pelletizado,
405             h,
406             d,
407         )
408     ]
409     == 0,
410     almacenamiento[(a1, Etapa.expedicion_harina, h, d)]
411     == 0,
412 )
413 model.add_if_then(
414     alm_libre[(a1, Etapa.embolsado_harina, h, d)] == 0,
415     almacenamiento[(a2, Etapa.embolsado_harina, h, d)] == 0,
416 )
417 model.add_if_then(
418     alm_libre[(a1, Etapa.embolsado_harina, h, d)] == 0,
419     almacenamiento[
420         (
421             a2,
422             Etapa.embolsado_pelletizado,
423             h,
424             d,
425         )
426     ]
427     == 0,
428 )
429 model.add_if_then(
430     alm_libre[

```

```

431         (
432             a2,
433             Etapa.embolsado_pelletizado,
434             h,
435             d,
436         )
437     ]
438     == 0,
439     almacenamiento[
440         (
441             a1,
442             Etapa.embolsado_pelletizado,
443             h,
444             d,
445         )
446     ]
447     == 0,
448 )
449 model.add_if_then(
450     alm_libre[
451         (
452             a2,
453             Etapa.embolsado_pelletizado,
454             h,
455             d,
456         )
457     ]
458     == 0,
459     almacenamiento[(a1, Etapa.embolsado_harina, h, d)] == 0,
460 )
461
462 # mutex entre alimentos
463 for i in Etapa:
464     for h in horas:
465         for d in dias:
466             model.add_constraint(
467                 model.sum(prod_bool[(a, i, h, d)] for a in alimentos) <= 1
468             )
469
470 # penalizacion por cambio
471 horas_menos_ult = horas[:-1]
472 for a1 in alimentos:
473     for a2 in alimentos:
474         for h in horas_menos_ult:
475             for d in dias:
476                 if a1 != a2:
477                     model.add_constraint(
478                         prod_bool[(a1, Etapa.etapa_0, h, d)]
479                         + prod_bool[(a2, Etapa.etapa_0, h + 1, d)]
480                         <= 1
481                     )
482
483
484 #####
485 # Solución
486 #####
487 solucion = model.solve()

```

```

488 solucion.display(iter_vars=[])
489 print(f"Took: {solucion.solve_details.time}")
490
491
492 #####
493 # Visualizaciones (gráficas)
494 #####
495 tiempo = list(product(dias, horas))
496 # tiempo = tiempo[:26] # Para las graficas chicas
497 x_values = list(map(lambda x: f"{x[0]}|{x[1]}", tiempo))
498 etapas = tipo_alimento_etapa.values()
499
500 def get_etapa_final(id_alimento):
501     return tipo_alimento_etapa[demanda[id_alimento][0]]
502
503 # Producción (etapas finales)
504 plt.figure(figsize=(12, 3))
505 plt.margins(0, 1)
506 bottom = [0] * len(tiempo)
507 for a in alimentos:
508     valores = [
509         produccion[(a, get_etapa_final(a), h, d)].solution_value
510         for (d, h) in tiempo
511     ]
512     plt.bar(x_values, valores, bottom=bottom, label=f"Alimento {a}")
513     bottom = [sum(x) for x in zip(bottom, valores)]
514 plt.legend()
515 plt.title("Producción (etapas finales)")
516
517 # Producción (etapa 0)
518 plt.figure(figsize=(12, 3))
519 plt.margins(0, 1)
520 bottom = [0] * len(tiempo)
521 for a in alimentos:
522     valores = [
523         produccion[(a, Etapa.etapa_0, h, d)].solution_value for (d, h) in tiempo
524     ]
525     plt.bar(x_values, valores, bottom=bottom, label=f"Alimento {a}")
526     bottom = [sum(x) for x in zip(bottom, valores)]
527 plt.legend()
528 plt.title("Producción (etapa 0)")
529
530 # Etapas activas
531 plt.figure(figsize=(12, 3))
532 plt.margins(0, 1)
533 bottom = [0] * len(dias)
534 for i in EtapaReal:
535     valores = list(
536         map(
537             lambda x: int(bool(x)),
538             [empleados_activos[(i, d)].solution_value for d in dias],
539         )
540     )
541     plt.bar(dias, valores, bottom=bottom, label=i.value)
542     bottom = [sum(x) for x in zip(bottom, valores)]
543 plt.legend()
544 plt.title("Etapas activas")

```

```

545
546 # Ingreso a almacenamiento
547 etapas = capacidad_almacenamiento.keys()
548 plt.figure(figsize=(12, 3))
549 plt.margins(0, 1)
550 bottom = [0] * len(tiempo)
551 for a in alimentos:
552     valores = [
553         sum([almacenamiento[(a, get_etapa_final(a), h, d)].solution_value]
554             for (d, h) in tiempo
555             ]
556     plt.bar(x_values, valores, bottom=bottom, label=f"Alimento {a}")
557     bottom = [sum(x) for x in zip(bottom, valores)]
558 plt.legend()
559 plt.title("Ingreso a almacenamiento")
560
561 # Alimento disponible en almacenamiento
562 plt.figure(figsize=(12, 3))
563 plt.margins(0, 1)
564 bottom = [0] * len(tiempo)
565 for a in alimentos:
566     valores = [
567         sum(
568             [
569                 get_alimento_disponible(
570                     a, get_etapa_final(a), h, d
571                 ).solution_value
572             ]
573         )
574         for (d, h) in tiempo
575     ]
576     plt.bar(x_values, valores, bottom=bottom, label=f"Alimento {a}")
577     bottom = [sum(x) for x in zip(bottom, valores)]
578 plt.legend()
579 plt.title("Alimento disponible en almacenamiento")
580
581 # Tabla con los resultados (para el anexo)
582 print("\\begin{longtable}[H]{|c|c|c|c|c|c|}")
583 print("\\hline")
584 print(
585     "\\textbf{Alimento} & \\textbf{Etapa} & \\textbf{Hora} & "
586     "\\textbf{Día} & \\textbf{Producción} & \\textbf{Almacen}"
587     "amiento} \\\\ \\hline"
588 )
589 for x in prod_range:
590     prod_value = produccion[x].solution_value
591     alm_value = almacenamiento[x].solution_value if x in alm_range else 0
592     if prod_value > 0 or alm_value > 0:
593         print(
594             f"{x[0]} & {x[1]} & {x[2] / GRANULARIDAD_HORA} & {x[3]} & "
595             f"{prod_value} & {alm_value} \\\\ \\hline"
596         )
597 print(
598     "\\caption{Resultados prueba X - modelado final (omitiendo valores nulos).}"
599 )
600 print("\\end{longtable}")

```


Anexo B

Resultados de las pruebas de correctitud

B.1. Pruebas de correctitud del primer modelo

Prueba 2:

Alimento	Etapa	Hora	Día	Producción
1	1	1	1	12.0
1	1	2	1	30.0
1	1	4	1	2.0
1	1	5	1	30.0
1	1	7	1	6.0
1	2	1	2	12.0
1	2	2	1	12.0
1	2	3	1	12.0
1	2	4	1	12.0
1	2	5	1	8.0
1	2	6	1	12.0
1	2	7	1	12.0
1	3	1	2	12.0
1	3	2	2	12.0
1	3	3	1	12.0
1	3	4	1	12.0
1	3	5	1	12.0
1	3	6	1	8.0
1	3	7	1	12.0
1	4	1	2	12.0
1	4	2	2	12.0
1	4	3	2	12.0

1	4	4	1	8.0
1	4	5	1	12.0
1	4	6	1	12.0
1	4	7	1	12.0
1	5	1	2	15.0
1	5	2	2	15.0
1	5	3	2	15.0
1	5	4	2	15.0
1	5	5	1	8.0
1	5	6	1	12.0

Tabla B.1: Resultados prueba 2 - primer modelado (omitiendo valores nulos).

Prueba 3:

Alimento	Etapa	Hora	Día	Producción
1	1	3	1	12.0
1	1	4	1	8.0
1	2	4	1	12.0
1	2	5	1	8.0
1	3	5	1	12.0
1	3	6	1	8.0
1	4	6	1	12.0
1	4	7	1	8.0
1	5	1	2	15.0
1	5	7	1	5.0
2	1	4	1	4.0
2	1	5	1	12.0
2	1	6	1	12.0
2	1	7	1	12.0
2	2	1	2	12.0
2	2	5	1	4.0
2	2	6	1	12.0
2	2	7	1	12.0
2	3	1	2	12.0
2	3	2	2	12.0
2	3	6	1	4.0
2	3	7	1	12.0
2	4	1	2	12.0
2	4	2	2	12.0
2	4	3	2	12.0
2	4	7	1	4.0
2	5	2	2	10.0
2	5	3	2	15.0
2	5	4	2	15.0

Tabla B.2: Resultados prueba 3 - primer modelado (omitiendo valores nulos).

B.2. Pruebas de correctitud del segundo modelo

Prueba 1:

Etapa	Hora	Día	Producción	Almacenamiento
Etapa 0	1	2	12.0	0
Etapa 0	2	2	12.0	0
Etapa 0	4	2	6.0	0
Expedición harina	1	2	10.0	12.0
Expedición harina	2	2	14.0	12.0
Expedición harina	4	2	6.0	6.0

Tabla B.3: Resultados prueba 1 - segundo modelado (omitiendo valores nulos).

Prueba 2:

Etapa	Hora	Día	Producción	Almacenamiento
Etapa 0	4	2	8.0	0
Etapa 0	6	2	12.0	0
Expedición harina	4	2	0	8.0
Expedición harina	5	2	6.0	0
Expedición harina	6	2	14.0	12.0
Etapa 0	1	2	8.0	0
Etapa 0	5	2	12.0	0
Embolsado harina	1	2	6.0	8.0
Embolsado harina	5	2	7.0	12.0
Embolsado harina	6	2	7.0	0

Tabla B.4: Resultados prueba 2 - segundo modelado (omitiendo valores nulos).

Prueba 3:

Etapa	Hora	Día	Producción	Almacenamiento
Etapa 0	1	2	12.0	0
Etapa 0	2	2	8.0	0
Expedición harina	1	2	12.0	12.0
Expedición harina	2	2	8.0	8.0
Etapa 0	3	2	10.0	0
Etapa 0	4	2	10.0	0
Pelletizado	3	2	10.0	0
Pelletizado	4	2	10.0	0
Expedición pelletizado	3	2	6.0	10.0
Expedición pelletizado	4	2	14.0	10.0

Tabla B.5: Resultados prueba 3 - segundo modelado (omitiendo valores nulos).

Prueba 4:

Etapa	Hora	Día	Producción	Almacenamiento
Etapa 0	5	1	10.0	0
Etapa 0	6	1	10.0	0
Pelletizado	5	1	10.0	0
Pelletizado	6	1	10.0	0
Expedición pelletizado	5	1	6.0	10.0
Expedición pelletizado	6	1	14.0	10.0
Etapa 0	1	1	10.0	0
Etapa 0	7	1	10.0	0
Pelletizado	1	1	10.0	0
Pelletizado	7	1	10.0	0
Embolsado pelletizado	1	1	0	10.0
Embolsado pelletizado	2	2	6.0	0
Embolsado pelletizado	3	2	7.0	0
Embolsado pelletizado	4	2	7.0	0
Embolsado pelletizado	7	1	0	10.0

Tabla B.6: Resultados prueba 4 - segundo modelado (omitiendo valores nulos).

Prueba 5:

Etapa	Hora	Día	Producción	Almacenamiento
Etapa 0	2	1	12.0	0
Etapa 0	3	1	9.0	0
Etapa 0	4	1	12.0	0
Etapa 0	6	1	12.0	0
Expedición harina	2	1	5.0	12.0
Expedición harina	3	1	14.0	9.0
Expedición harina	4	1	14.0	12.0
Expedición harina	6	1	12.0	12.0
Etapa 0	3	2	8.0	0
Etapa 0	4	2	12.0	0
Embolsado harina	3	2	7.0	8.0
Embolsado harina	4	2	7.0	12.0
Embolsado harina	5	2	6.0	0
Etapa 0	1	1	10.0	0
Etapa 0	5	1	10.0	0
Etapa 0	7	1	10.0	0
Pelletizado	1	1	10.0	0
Pelletizado	5	1	10.0	0
Pelletizado	7	1	10.0	0
Expedición pelletizado	1	1	10.0	10.0
Expedición pelletizado	5	1	10.0	10.0
Expedición pelletizado	7	1	10.0	10.0
Etapa 0	1	2	7.0	0
Etapa 0	2	2	7.0	0
Pelletizado	1	2	7.0	0
Pelletizado	2	2	7.0	0
Embolsado pelletizado	1	2	7.0	7.0
Embolsado pelletizado	2	2	7.0	7.0

Tabla B.7: Resultados prueba 5 - segundo modelado (omitiendo valores nulos).

B.3. Pruebas de correctitud del modelado final

Prueba 1:

Alimento	Etapa	Hora	Día	Producción	Almacenamiento
1	Etapa 0	0.5	2	6.0	0
1	Etapa 0	3.5	1	6.0	0
1	Etapa 0	4.0	1	6.0	0
1	Etapa 0	4.5	1	6.0	0
1	Etapa 0	5.0	1	6.0	0
1	Etapa 0	5.5	1	6.0	0
1	Etapa 0	6.0	1	6.0	0
1	Etapa 0	6.5	1	6.0	0
1	Etapa 0	7.0	1	6.0	0
1	Expedición harina	0.5	2	7.0	6.0
1	Expedición harina	3.5	1	0	6.0
1	Expedición harina	4.0	1	5.0	6.0
1	Expedición harina	4.5	1	7.0	6.0
1	Expedición harina	5.0	1	7.0	6.0
1	Expedición harina	5.5	1	7.0	6.0
1	Expedición harina	6.0	1	7.0	6.0
1	Expedición harina	6.5	1	7.0	6.0
1	Expedición harina	7.0	1	7.0	6.0
2	Etapa 0	0.5	1	6.0	0
2	Etapa 0	1.0	1	6.0	0
2	Etapa 0	1.5	1	6.0	0
2	Etapa 0	2.0	1	6.0	0
2	Etapa 0	2.5	1	6.0	0
2	Embolsado harina	0.5	1	0	6.0
2	Embolsado harina	1.0	1	0	6.0
2	Embolsado harina	1.5	1	0	6.0
2	Embolsado harina	2.0	1	0	6.0
2	Embolsado harina	2.5	1	0	6.0
2	Embolsado harina	3.0	1	2.0	0
2	Embolsado harina	3.5	1	3.5	0
2	Embolsado harina	4.0	1	3.5	0
2	Embolsado harina	4.5	1	3.5	0
2	Embolsado harina	5.0	1	3.5	0
2	Embolsado harina	5.5	1	3.5	0
2	Embolsado harina	6.0	1	3.5	0
2	Embolsado harina	6.5	1	3.5	0
2	Embolsado harina	7.0	1	3.5	0

Tabla B.8: Resultados prueba 1 - modelado final (omitiendo valores nulos).

Prueba 2:

Alimento	Etapa	Hora	Día	Producción	Almacenamiento
1	Etapa 0	0.5	1	6.0	0
1	Etapa 0	1.0	1	6.0	0
1	Etapa 0	1.5	1	6.0	0
1	Etapa 0	2.0	1	6.0	0

1	Etapa 0	2.5	1	6.0	0
1	Expedición harina	0.5	1	2.0	6.0
1	Expedición harina	1.0	1	0	6.0
1	Expedición harina	1.5	1	7.0	6.0
1	Expedición harina	2.0	1	7.0	6.0
1	Expedición harina	2.5	1	7.0	6.0
1	Expedición harina	3.0	1	7.0	0
2	Etapa 0	3.5	1	6.0	0
2	Etapa 0	4.0	1	6.0	0
2	Etapa 0	4.5	1	6.0	0
2	Etapa 0	5.0	1	6.0	0
2	Etapa 0	5.5	1	6.0	0
2	Expedición harina	3.5	1	2.0	6.0
2	Expedición harina	4.0	1	7.0	6.0
2	Expedición harina	4.5	1	7.0	6.0
2	Expedición harina	5.0	1	7.0	6.0
2	Expedición harina	5.5	1	7.0	6.0
3	Etapa 0	6.5	1	6.0	0
3	Etapa 0	7.0	1	6.0	0
3	Expedición harina	6.5	1	5.0	6.0
3	Expedición harina	7.0	1	7.0	6.0

Tabla B.9: Resultados prueba 2 - modelado final (omitiendo valores nulos).

Prueba 3:

Alimento	Etapa	Hora	Día	Producción	Almacenamiento
1	Etapa 0	0.5	1	6.0	0
1	Etapa 0	4.5	1	6.0	0
1	Etapa 0	5.0	1	6.0	0
1	Etapa 0	5.5	1	6.0	0
1	Etapa 0	6.0	1	6.0	0
1	Etapa 0	6.5	1	6.0	0
1	Etapa 0	7.0	1	4.0	0
1	Expedición harina	0.5	1	0	6.0
1	Expedición harina	4.0	1	5.0	0
1	Expedición harina	4.5	1	0	6.0
1	Expedición harina	5.0	1	7.0	6.0
1	Expedición harina	5.5	1	7.0	6.0
1	Expedición harina	6.0	1	7.0	6.0
1	Expedición harina	6.5	1	7.0	6.0
1	Expedición harina	7.0	1	7.0	4.0
2	Etapa 0	1.5	1	6.0	0
2	Etapa 0	2.0	1	6.0	0
2	Etapa 0	2.5	1	6.0	0
2	Etapa 0	3.0	1	6.0	0
2	Etapa 0	3.5	1	6.0	0
2	Embolsado harina	0.5	2	3.5	0
2	Embolsado harina	1.0	2	2.0	0
2	Embolsado harina	1.5	1	0	6.0
2	Embolsado harina	1.5	2	3.5	0
2	Embolsado harina	2.0	1	0	6.0

2	Embolsado harina	2.0	2	3.5	0
2	Embolsado harina	2.5	1	0	6.0
2	Embolsado harina	2.5	2	3.5	0
2	Embolsado harina	3.0	1	0	6.0
2	Embolsado harina	3.0	2	3.5	0
2	Embolsado harina	3.5	1	0	6.0
2	Embolsado harina	3.5	2	3.5	0
2	Embolsado harina	4.0	2	3.5	0
2	Embolsado harina	4.5	2	3.5	0
3	Etapa 0	0.5	2	5.0	0
3	Etapa 0	1.0	2	5.0	0
3	Etapa 0	1.5	2	5.0	0
3	Etapa 0	2.0	2	5.0	0
3	Etapa 0	2.5	2	5.0	0
3	Etapa 0	3.0	2	5.0	0
3	Etapa 0	3.5	2	5.0	0
3	Etapa 0	4.0	2	5.0	0
3	Pelletizado	0.5	2	5.0	0
3	Pelletizado	1.0	2	5.0	0
3	Pelletizado	1.5	2	5.0	0
3	Pelletizado	2.0	2	5.0	0
3	Pelletizado	2.5	2	5.0	0
3	Pelletizado	3.0	2	5.0	0
3	Pelletizado	3.5	2	5.0	0
3	Pelletizado	4.0	2	5.0	0
3	Expedición pelletizado	0.5	2	0	5.0
3	Expedición pelletizado	1.0	2	7.0	5.0
3	Expedición pelletizado	1.5	2	5.0	5.0
3	Expedición pelletizado	2.0	2	7.0	5.0
3	Expedición pelletizado	2.5	2	0	5.0
3	Expedición pelletizado	3.0	2	0	5.0
3	Expedición pelletizado	3.5	2	7.0	5.0
3	Expedición pelletizado	4.0	2	7.0	5.0
3	Expedición pelletizado	4.5	2	7.0	0
4	Etapa 0	0.5	3	5.0	0
4	Etapa 0	1.5	3	5.0	0
4	Etapa 0	2.0	3	5.0	0
4	Etapa 0	5.0	2	5.0	0
4	Etapa 0	5.5	2	5.0	0
4	Etapa 0	6.0	2	5.0	0
4	Pelletizado	0.5	3	5.0	0
4	Pelletizado	1.5	3	5.0	0
4	Pelletizado	2.0	3	5.0	0
4	Pelletizado	5.0	2	5.0	0
4	Pelletizado	5.5	2	5.0	0
4	Pelletizado	6.0	2	5.0	0
4	Embolsado pelletizado	0.5	3	3.5	5.0
4	Embolsado pelletizado	1.0	3	3.5	0
4	Embolsado pelletizado	1.5	3	3.5	5.0
4	Embolsado pelletizado	2.0	3	3.5	5.0
4	Embolsado pelletizado	2.5	3	3.5	0
4	Embolsado pelletizado	5.0	2	3.5	5.0
4	Embolsado pelletizado	5.5	2	2.0	5.0
4	Embolsado pelletizado	6.0	2	0	5.0

4	Embolsado pelletizado	6.5	2	3.5	0
4	Embolsado pelletizado	7.0	2	3.5	0
5	Etapa 0	1.5	4	6.0	0
5	Etapa 0	2.0	4	6.0	0
5	Etapa 0	6.0	4	6.0	0
5	Etapa 0	7.0	2	2.0	0
5	Expedición harina	1.5	4	6.0	6.0
5	Expedición harina	2.0	4	6.0	6.0
5	Expedición harina	6.0	4	6.0	6.0
5	Expedición harina	7.0	2	2.0	2.0
6	Etapa 0	0.5	4	5.0	0
6	Etapa 0	3.0	4	5.0	0
6	Etapa 0	3.5	4	5.0	0
6	Etapa 0	4.0	4	5.0	0
6	Etapa 0	4.5	4	5.0	0
6	Etapa 0	7.0	4	5.0	0
6	Pelletizado	0.5	4	5.0	0
6	Pelletizado	3.0	4	5.0	0
6	Pelletizado	3.5	4	5.0	0
6	Pelletizado	4.0	4	5.0	0
6	Pelletizado	4.5	4	5.0	0
6	Pelletizado	7.0	4	5.0	0
6	Expedición pelletizado	0.5	4	5.0	5.0
6	Expedición pelletizado	3.0	4	0	5.0
6	Expedición pelletizado	3.5	4	6.0	5.0
6	Expedición pelletizado	4.0	4	7.0	5.0
6	Expedición pelletizado	4.5	4	7.0	5.0
6	Expedición pelletizado	7.0	4	5.0	5.0
7	Etapa 0	3.0	3	3.5	0
7	Etapa 0	3.5	3	5.5	0
7	Etapa 0	4.5	3	6.0	0
7	Etapa 0	5.0	3	6.0	0
7	Etapa 0	6.0	3	6.0	0
7	Etapa 0	7.0	3	3.0	0
7	Embolsado harina	3.0	3	3.5	3.5
7	Embolsado harina	3.5	3	3.5	5.5
7	Embolsado harina	4.0	3	2.0	0
7	Embolsado harina	4.5	3	3.5	6.0
7	Embolsado harina	5.0	3	3.5	6.0
7	Embolsado harina	5.5	3	3.5	0
7	Embolsado harina	6.0	3	3.5	6.0
7	Embolsado harina	6.5	3	3.5	0
7	Embolsado harina	7.0	3	3.5	3.0

Tabla B.10: Resultados prueba 3 - modelado final (omitiendo valores nulos).