



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
MÁSTER DE INGENIERÍA INDUSTRIAL**

**MODELOS ANALÍTICOS E INTELIGENCIA ARTIFICIAL
PARA LA MEJORA DE OPERACIONES**

Autor: García Ponce, Miguel

Director: Cruz Sánchez de Rojas, Álvaro De La

Coordinador: Ortiz Marcos, Susana

Madrid, Junio 2018

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. *Miguel García Ponce* DECLARA ser el titular de los derechos de propiedad intelectual de la obra: *Modelos Analíticos e Inteligencia Artificial para la mejora de operaciones*, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor CEDE a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 21 de Junio de 2018

ACEPTA

Fdo Miguel García Ponce

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Modelos Analíticos e Inteligencia Artificial para la mejora de operaciones

.....
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico2017 / 18..... es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.

Fdo.:



Miguel García Ponce

Fecha: ..21../ ..06../ 2018

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.:



Fecha: 21 / 06 / 2018

ÁLVARO DE LA CRUZ SÁNCHEZ



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
MÁSTER DE INGENIERÍA INDUSTRIAL**

**MODELOS ANALÍTICOS E INTELIGENCIA ARTIFICIAL
PARA LA MEJORA DE OPERACIONES**

Autor: García Ponce, Miguel

Director: Cruz Sánchez de Rojas, Álvaro De La

Coordinador: Ortiz Marcos, Susana

Madrid, Junio 2018

MODELOS ANALÍTICOS E INTELIGENCIA ARTIFICIAL PARA LA MEJORA DE OPERACIONES

Autor: García Ponce, Miguel.

Director: Cruz Sánchez de Rojas, Álvaro De La.

Entidad Colaboradora: PricewaterhouseCoopers Asesores de Negocio S.L.

RESUMEN DEL PROYECTO

1. Introducción

El sector Retail ha experimentado en los últimos años una serie de cambios que han modificado significativamente su estructura y su operativa. Estos cambios abarcan no sólo la configuración del propio mercado, sino que también el propio consumidor ha modificado su forma de actuar. Los factores principales que lo han promovido se pueden agrupar en tres: globalización, el aumento de la visibilidad dentro del propio mercado y los avances tecnológicos. Este hecho ha propiciado la necesidad de adaptarse que tiene el sector de cara a sus clientes y, por lo tanto, que el propio consumidor se haya convertido en el núcleo principal del mercado, cuyas necesidades hay que satisfacer.

Como consecuencia de estos hechos, los profesionales del sector buscan en la actualidad desarrollar nuevas técnicas y estructuras que permitan ser implementadas tanto a nivel físico como a nivel de software con el fin de optimizar sus procesos y mejorar, principalmente, su eficiencia.

Las soluciones que se plantean en la actualidad por parte de las empresas se pueden dividir, por un lado, en soluciones físicas como la reestructuración de las redes de distribución y logística y, por otro lado, con la implementación de nuevos softwares y modelos que permiten la gestión eficiente de dichas redes. Además, a día de hoy existe una gran necesidad por adelantarse al comportamiento del mercado y el cliente con el objetivo de establecer estrategias y lanzar nuevas técnicas y productos que permitan diferenciarse. Por ello, el desarrollo de modelos de predicción fiables y robustos es una tarea pendiente que poco a poco las empresas están comenzando a instaurar y desarrollar en sus entornos de negocio. Claros ejemplos de esta necesidad son los casos de Alibaba [WANG17] y Mercadona, donde, por un lado, el primero ha mostrado ya grandes resultados gracias a la implementación de soluciones avanzadas relacionadas con las necesidades del mercado y, por otro lado, Mercadona muestra esa necesidad de desarrollo de dichas técnicas con las palabras de su CEO, Juan Roig [MERI17].

Por todas estas razones mencionadas, el objetivo de este proyecto es desarrollar dos modelos complementarios relacionados con los problemas existentes del sector mencionados, que permitan la optimización de la red de distribución a través de un algoritmo de mejora de la gestión de la misma. Además, el desarrollo de un modelo de predicción que facilite a los

negocios adelantarse a las necesidades pudiendo, a la vez, optimizar sus procesos de abastecimiento y de cobertura de demanda con el fin de aumentar la eficiencia a la par que se disminuyen los costes.

2. Metodología

El procedimiento llevado a cabo en la realización de este proyecto se ha aplicado de la misma manera en el desarrollado de los dos modelos explicados debido a que ambos están basados en técnicas de minería de datos y de desarrollo de algoritmos de optimización.

En primer lugar, la recopilación de información del mercado es fundamental de cara a la posterior generación de datos. Estos datos serán empleados con el fin de desarrollar un negocio con las características estándar del sector. Dicho de otra manera, se ha llevado a cabo el diseño de toda la base de datos de un negocio tipo donde se han definido todos los componentes que lo caracterizan a través de una base de datos relacional.

Para el caso del modelo de optimización, se ha desarrollado una prueba de concepto en la que se han construido una serie de algoritmos que permiten la generación de los pedidos, devoluciones y stock inicial para un período de tiempo determinado. El siguiente paso, una vez obtenidos todos los datos de entrada del modelo, es el desarrollo de la lógica de la herramienta en cuestión, denominada OMS (*Order Management System*). Posteriormente, esta herramienta será aplicada y adaptada a cuatro escenarios posibles que caracterizan diversas configuraciones de la red de distribución del negocio, permitiendo evaluar el comportamiento y los costes de cada una de ellos.

Para el modelo de predicción, se proponen inicialmente dos tipos posibles de técnicas de predicción basadas en fundamentos de aprendizaje supervisado (*Machine Learning*), ARIMA y FbProphet. Para ambos casos, se llevará a cabo el desarrollo de la lógica de programación, la correspondiente evaluación de los resultados y la optimización de cada uno de ellos. Finalmente, se decidirá en base a los resultados obtenidos, qué modelo es más preciso y cuál se adapta mejor a las necesidades del negocio objeto del proyecto.

Cabe destacar que los procedimientos seguidos y los datos empleados en la definición de los modelos han sido construidos en base a datos y técnicas reales del mercado, con el fin de aportar un grado de realismo y desarrollar una herramienta aplicable a cualquier empresa del sector.

3. Resultados

3.1. Modelo de optimización (OMS)

En primer lugar, para la definición de la base de datos se requiere el desarrollo de un algoritmo de generación aleatoria desarrollado para obtener los datos maestros de los pedidos, las devoluciones y el stock inicial, elementos fundamentales del modelo de optimización.

A nivel general, la red de distribución está formada por un centro de almacenamiento y un conjunto de 14 tiendas repartidas a lo largo del territorio nacional. A parte de las tiendas físicas, también se ha definido la tienda online con el objetivo de introducir el concepto de *e-commerce*. En cuanto a la base de datos, hay un total de 10 maestros, divididos en aquellos que caracterizan la red de distribución (tiendas, almacenes y clientes) y aquellos que definen el inventario del negocio definido (artículos, inventario, pedidos, etc.).

La definición de los escenarios, no sólo modifica la estructura de la red de distribución, sino que también añade conceptos propios y significativos del sector. A continuación, se resume cada uno:

- **Escenario 1.** Modelo centralizado donde los pedidos de la tienda online se abastecen con el inventario del almacén, mientras que las tiendas físicas se autoabastecen dando lugar a una disgregación entre ambas.
- **Escenario 2.** La gestión de la tienda física se mantiene, mientras que los pedidos online pueden ser abastecidos con el inventario tanto del almacén como el de las propias tiendas.
- **Escenario 3.** La estructura y lógica de este escenario es la misma que en el anterior, sin embargo se añade el concepto de stock de seguridad como medida a los resultados obtenidos.
- **Escenario 4.** Contiene todos los conceptos hasta ahora añadidos, además de la incorporación del servicio intertienda.

Además, se han establecido una serie de costes que serán imputados dentro del modelo y que permitirán cuantificar las actividades significativas que involucra la gestión de un pedido dentro de la cadena de suministro de una compañía . A continuación, se presentan los resultados:

	E1	E2	E3	E4
Ingresos en tienda	1.477.581,23 €	1.446.548,54 €	1.470.058,17 €	1.473.902,57 €
Ingresos online	344.532,59 €	383.036,54 €	383.036,84 €	383.036,84 €
Ingresos totales	1.822.113,82 €	1.829.585,08 €	1.853.095,01 €	1.856.939,41 €
Coste Mano de Obra Tienda	27.387,00 €	34.718,00 €	34.874,00 €	34.945,00 €
Coste Mano de Obra Almacén	5.486,00 €	213,00 €	234,00 €	234,00 €
Coste Packaging Tienda	- €	16.351,00 €	16.351,00 €	16.714,00 €
Coste Packaging Almacén	15.541,00 €	603,00 €	603,00 €	603,00 €
Coste Transporte Tienda	- €	390.646,00 €	478.768,00 €	484.937,00 €
Coste Transporte Almacén	993.269,00 €	15.698,00 €	21.353,78 €	21.353,78 €
Coste Rotura Stock Tienda	136.785,60 €	170.121,99 €	145.021,80 €	141.347,34 €
Coste Rotura Stock Almacén	34.833,60 €	- €	- €	- €
Coste de Venta	1.213.302,20 €	628.350,99 €	697.205,80 €	699.953,34 €
Coste de Mercancías	728.845,53 €	731.834,03 €	741.238,00 €	742.775,76 €
Coste Total	1.942.147,73 €	1.360.185,02 €	1.438.443,80 €	1.442.729,10 €
Margen Bruto	-120.033,91 €	469.400,06 €	414.651,21 €	414.210,31 €

Unidades servidas en tienda origen	15386	15038	15302	15294
Unidades servidas en otra tienda (intertienda)	0	0	0	47
Pedidos servidos en tienda	0	4119	4103	4103
Pedidos servidos en almacén	3917	167	183	183
Unidades servidas totales	19303	19324	19588	19627
Precio medio unidad servida	94,40 €	94,67 €	94,60 €	94,61 €
Venta en tienda no servida (OMS)	0	348	84	45
Venta en tienda no servida (estimación)	1449	1449	1449	1449
Pedidos online no servidos	369	0	0	0
Unidades totales no servidas	1818	1797	1533	1494
Rotura de stock	9,24%	9,13%	7,79%	7,59%

En resumen, el escenario que representa el sistema centralizado muestra unos resultados bastante negativos tanto a nivel económico como de gestión, motivo por el cual las empresas del sector están en proceso de reestructuración o ya han cambiado su estructura. Sin embargo, los otros tres escenarios mejoran los resultados conforme se van añadiendo características al modelo. Por lo tanto, el Escenario 4 sería el más acertado ya que a nivel económico no presenta grandes diferencias respecto a los demás, mientras que en relación a la eficiencia y servicio prestado al cliente es el que mejores valores presenta.

3.2. Modelo de predicción

Al igual que en el modelo de optimización, en este también es necesario el desarrollo de un algoritmo de generación para poder obtener un histórico de datos en el cual se represente el desglose de ventas diarias correspondiente a tres temporadas de invierno consecutivas para el negocio tipo diseñado. En ese set de datos generado se tienen en cuenta factores como el clima y los días de vacaciones y promociones. Estos tienen gran influencia en el comportamiento del modelo ya que aportan realismo y variaciones puntuales de la línea de tendencia, que son aquellas que interesan al negocio de cara a optimizar su gestión. Todos estos datos se han obtenido de fuentes fiables y empleadas en la realidad como son Aemet y Google Trends. Posteriormente, estos datos serán tratados y analizados con objeto de ver la interacción entre ellos, llevando a cabo una parametrización que permita caracterizar la lógica de generación del histórico.

Los dos modelos de predicción aplicados al caso de negocio son ARIMA y Prophet. Ambos están orientados al desarrollo de pronósticos sobre series temporales mediante la aplicación de técnicas de regresión lineal, aunque cada uno presenta propiedades y parámetros de modelización propios.

Con el fin de abordar el desarrollo de los modelos propuestos, se ha seguido la metodología estándar de resolución típica del sector. En primer lugar, se ha generado el set de datos de entrada y se ha tratado de acuerdo con las especificaciones dictadas por los profesionales de dicha materia, dividiendo el set de datos en dos partes, entrenamiento y validación. Para ambos modelos, se ha llevado a cabo un análisis de sensibilidad de los parámetros de entrada de cada una de las funciones internas con el objetivo de encontrar aquellos valores que dan lugar a el

resultado óptimo para el caso de negocio en cuestión. Una vez llevado a cabo el proceso de optimización, se procede a comparar los resultados de ambos modelos haciendo uso de unos indicadores estadísticos propios de los proyectos de la minería de datos. Estos indicadores miden el error entre la predicción obtenida y los valores reales de testeo.

Modelo	MAPE	MAE	MSE
ARIMA	5,59	19,01	50,87
FbProphet	4,10	10,62	29,31

A la vista de los resultados obtenidos, se puede afirmar con claridad que el modelo que mejor respuesta tiene es el desarrollado por Facebook, FbProphet. Los tres errores toman valores bastante inferiores con respecto a los del modelo ARIMA, lo cual significa que el ajuste de la línea de predicción es más preciso para el caso del modelo desarrollado por Facebook.

A parte de los errores, se considera que este modelo es más completo debido a que tiene en cuenta una serie de aspectos que permiten añadir los dos factores que desde el principio eran objetivo del proyecto, el clima y las promociones, proveyendo unos resultados con mayor nivel de realidad.

4. Conclusiones

Tras la realización de este proyecto, se ha observa la necesidad imperiosa que tienen las empresas del sector por instaurar este tipo de herramientas en su día a día. Pese a que suponen una fuerte inversión inicial y la contratación de profesionales especializados en sector, la repercusión tanto a nivel económico como de gestión resulta ser muy significativa.

El desarrollo de un modelo de predicción de demanda permite a las empresas tener un conocimiento previo del comportamiento del mercado y de sus clientes frente a las fluctuaciones propias de cada uno de ellos. Ser conocedor de esta información aporta una ventaja de cara al futuro proceso de abastecimiento y de fijación de precios con el fin de obtener el mayor margen de beneficios posible.

El modelo de predicción es un complemento perfecto para el sistema de optimización OMS desarrollado ya que facilita a las empresas los procesos de reestructuración de su red de distribución y la gestión eficiente de su inventario y posterior cobertura de demanda. Por tanto, como consecuencia de la implantación de estos sistemas lo que se obtiene es un ahorro de costes a la vez que se mejora en eficiencia y, por lo tanto, en la imagen dada a los clientes.

5. Referencias

- [WANG17] Wang, H. H. (12 de 11 de 2017). *Forbes*. Obtenido de Forbes:
<https://www.forbes.com/sites/helenwang/2017/11/12/alibabas-singles-day-by-the-numbers-a-record-25-billion-haul/#5f0ebe141db1>
- [MERI17] Merino, P. P. (3 de 3 de 2017). *E-commercenews*. Obtenido de E-commercenews:
<https://ecommerce-news.es/juan-roig-mercadona-nuestra-web-actual-una-mierda-56155>

ANALYTICAL MODELS AND ARTIFICIAL INTELLIGENCE FOR OPERATIONS IMPROVEMENT

Author: García Ponce, Miguel.

Mentor: Cruz Sánchez de Rojas, Álvaro De La.

Colaborative Entity: PricewaterhouseCoopers Asesores de Negocio S.L.

ABSTRACT

1. Introduction

The retail and consumer sector has experienced a series of changes in recent years that have significantly modified its structure and operations. These changes include not only the configuration of the market itself but also the consumer himself has modified his way of acting. The main factors that have promoted this change can be grouped into three: globalization, the increase in visibility within the market itself and technological advances. This has led to the need for the sector to adapt to its customers and, therefore, that the consumer himself has become the main core of the market, whose needs must be met.

As a result of these facts, professionals in the sector are currently looking to develop new techniques and structures that can be implemented both physically and at the software level in order to optimize their processes and improve, mainly, their efficiency.

The solutions that are currently being proposed by companies in the sector can be divided, on the one hand, into physical solutions such as the restructuring of distribution and logistics networks and, on the other hand, with the implementation of new software and models that allow the efficient management of these networks. In addition, today there is a great need to anticipate the behavior of the market and the client in order to establish strategies and launch new techniques and products. Therefore, the development of reliable and robust prediction models is a pending task that little by little companies are beginning to establish and develop in their business environments. Clear examples of this need are the cases of Alibaba [WANG17] and Mercadona, where, on the one hand, the first has already shown great results thanks to the implementation of advanced solutions related to the needs of the market and, on the other hand, Mercadona shows that need for the development of these techniques [MERI17].

For all these reasons mentioned, the objective of this project is to develop two complementary models related to the existing problems of the aforementioned sector, which allow the optimization of the distribution through an algorithm to improve their management. In addition, the development of a forecast model that facilitates businesses to anticipate the needs while being able to optimize their supply processes and demand coverage in order to increase efficiency while reducing costs.

2. Methodology

The procedure carried out in the realization of this project has been applied in the same way in the developed of the two models explained because both are based on techniques of data mining and development of optimization algorithms.

First of all, the collection of information and data specific to the market is fundamental for the subsequent generation of data. These data will be used in order to develop a business with the standard characteristics of the market. In other words, it has been carried out the design of the entire database of a business type of the sector where all the components that characterize it have been defined.

In the case of the optimization model, a proof of concept has been developed in which a series of algorithms have been constructed that allow the generation of orders, returns and initial stock for a specific period of time. The next step, once all the input data of the model has been obtained, is the development of the logic of the tool in question, called OMS (Order Management System). Subsequently, this tool will be applied and adapted to four possible scenarios that characterize different configurations of the business distribution network, allowing to evaluate the behavior and costs of each one of them.

For the forecasting model, two possible types of prediction techniques based on fundamentals of supervised learning (Machine Learning), ARIMA and FbProphe. For both cases, the development of the programming logic will be carried out, the corresponding evaluation of the results and the optimization of each one of them. Finally, it will be decided based on the results obtained, which model is more precise and better adapted to the needs of the business object of the project.

It should be noted that the procedures followed and the data used in the definition of the models have been built on the basis of data and real market techniques, in order to provide a degree of realism and develop a tool applicable to any company in the sector.

3. Results

3.1. Modelo de optimización (OMS)

In the first place, the development of a random generation algorithm developed to obtain the master data of the orders, the returns and the initial stock, fundamental elements of the optimization model, is required for the definition of the database.

At a general level, the distribution network consists of a storage center and a set of 14 stores distributed throughout the national territory. Apart from physical stores, the online store has also been defined with the aim of introducing the concept of e-commerce. Regarding the database, there are a total of 10 master data, divided into those that characterize the distribution network (stores, warehouses and customers) and those that define the defined business inventory (items, inventory, orders, etc.).

The definition of the scenarios not only modifies the structure of the distribution network, but also adds own and significant concepts of the sector. Next, each one is summarized:

- **Scenario 1.** A Centralized model where the orders of the online store are supplied with the inventory of the warehouse, while the physical stores self-supply resulting in a disaggregation between both.
- **Scenario 2.** The management of the physical store is maintained, while online orders can be supplied with the inventory of both the store and the stores themselves.
- **Scenario 3.** The structure and logic of this scenario are the same as in the previous one, however the concept of security stock is added as a measure to the results obtained.
- **Scenario 4.** Contains all the concepts so far added, in addition to the incorporation of the intertienda service.

Apart from the scenarios, a series of costs have been established that will be imputed within the model and that will allow quantifying the significant activities that involve the management of an order within the supply chain of a company. The results are presented:

	E1	E2	E3	E4
Revenues in store	1.477.581,23 €	1.446.548,54 €	1.470.058,17 €	1.473.902,57 €
Revenues in warehouse	344.532,59 €	383.036,54 €	383.036,84 €	383.036,84 €
Total Revenues	1.822.113,82 €	1.829.585,08 €	1.853.095,01 €	1.856.939,41 €
Workforce Cost in store	27.387,00 €	34.718,00 €	34.874,00 €	34.945,00 €
Workforce Cost in warehouse	5.486,00 €	213,00 €	234,00 €	234,00 €
Packaging Cost in store	- €	16.351,00 €	16.351,00 €	16.714,00 €
Packaging Cost in warehouse	15.541,00 €	603,00 €	603,00 €	603,00 €
Transpor Cost in store	- €	390.646,00 €	478.768,00 €	484.937,00 €
Transpor Cost in warehouse	993.269,00 €	15.698,00 €	21.353,78 €	21.353,78 €
Breakage of Stock Cost in store	136.785,60 €	170.121,99 €	145.021,80 €	141.347,34 €
Breakage of Stock Cost in warehouse	34.833,60 €	- €	- €	- €
Cost of Sales	1.213.302,20 €	628.350,99 €	697.205,80 €	699.953,34 €
Cost of Goods	728.845,53 €	731.834,03 €	741.238,00 €	742.775,76 €
Total Cost	1.942.147,73 €	1.360.185,02 €	1.438.443,80 €	1.442.729,10 €
Gross Margin	-120.033,91 €	469.400,06 €	414.651,21 €	414.210,31 €
Units served in store	15386	15038	15302	15294
Units served from other store (intertienda)	0	0	0	47
Orders served in store	0	4119	4103	4103
Orders served in warehouse	3917	167	183	183
Total units served	19303	19324	19588	19627
Unit served average price	94,40 €	94,67 €	94,60 €	94,61 €
Orders not served in store (OMS)	0	348	84	45
Orders not served in store (estimation)	1449	1449	1449	1449
Online orders not served	369	0	0	0
Total units not served	1818	1797	1533	1494
Breakage of Stock	9,24%	9,13%	7,79%	7,59%

In summary, the scenario represented by the centralized system presents quite negative results both at an economic and management level, which is why companies in the sector are in the process of restructuring or have already changed their structure. However, the other three scenarios improve the results as characteristics are added to the model. Therefore, Scenario 4 would be the most successful since at the economic level you do not present large differences with respect to the others, while in relation to the efficiency and service provided to the client, it is the one with the best values.

3.2. Forecasting model

As in the optimization model, it is also necessary to develop a generation algorithm in order to obtain a data history in which the breakdown of daily sales corresponding to three consecutive winter seasons for the business type designed is represented. In this set of generated data factors such as weather and holidays and promotions days are taken into account. These have great influence on the behavior of the model as they provide realism and specific variations of the trend line, which are those that interest the business in order to optimize its management. All these data have been obtained from reliable sources and applied in reality such as Aemet and Google Trends. Subsequently, these data will be treated and analyzed in order to see the interaction between them, developing a parameterization that allows characterizing the logic of generation of the historical.

The two prediction models applied to the business case are ARIMA and Prophet. Both are oriented to the development of forecasts on time series through the application of linear regression techniques. However, each one has its own characteristics and modeling parameters.

In order to address the development of the proposed models, the standard resolution methodology typical of the sector has been followed. First, the input data set has been generated and treated in accordance with the specifications dictated by the professionals of the sector, dividing the data set into two parts, training and validation. For both models, a sensitivity analysis of the input parameters of each of the internal functions has been carried out in order to find those values that give rise to the optimal result for the business case in question. Once the optimization process has been carried out, the results of both models are compared using statistical indicators from the data mining projects. These indicators measure the error between the prediction obtained and the actual test values.

Models	MAPE	MAE	MSE
ARIMA	5,59	19,01	50,87
FbProphet	4,10	10,62	29,31

In view of the results obtained, it can be clearly stated that the model with the best response is the one developed by Facebook, FbProphet. The three errors take much lower values with

respect to those of the ARIMA model, which means that the adjustment of the prediction line is more precise for the case of the model developed by Facebook.

Apart from the errors, it is considered that this model is more complete because it takes into account a series of aspects that allow adding the two factors that from the beginning were the objective of the project, the weather and the promotions, providing results with a greater level of reality.

4. Conclusions

After the completion of this project, it has been observed the imperative need for companies in the sector to install this type of tools in their day to day. Although they represent a strong initial investment and the hiring of professionals specialized in the sector, the repercussion both economically and in management is very significant.

The development of a demand prediction model allows companies to have a prior knowledge of the behavior of the market and its customers against the fluctuations of each one of them. Being aware of this information provides an advantage for the future process of supply and pricing in order to obtain the greatest possible profit margin.

The prediction model is a perfect complement to the OMS optimization system developed since it facilitates companies the processes of restructuring their distribution network and the efficient management of their inventory and subsequent demand coverage. Therefore, as a result of the implementation of these systems what is obtained is a cost saving while improving efficiency and, therefore, in the image given to customers.

5. References

- [WANG17] Wang, H. H. (12 de 11 de 2017). *Forbes*. Obtenido de Forbes:
<https://www.forbes.com/sites/helenwang/2017/11/12/alibabas-singles-day-by-the-numbers-a-record-25-billion-haul/#5f0ebe141db1>
- [MERI17] Merino, P. P. (3 de 3 de 2017). *E-commercenews*. Obtenido de E-commercenews:
<https://ecommerce-news.es/juan-roig-mercadona-nuestra-web-actual-una-mierda-56155>

Índice de Contenidos

RESUMEN DEL PROYECTO	IX
ABSTRACT.....	XV
Índice de Figuras.....	XXIII
Índice de Tablas	XXV
Acrónimos y Términos	XXVI
1. Introducción	1
1.1. Contexto del sector.....	1
1.2. Motivación del proyecto	5
2. Estado del arte	7
2.1. Metodología actual en redes de distribución.....	7
2.2. Comportamiento del cliente	8
2.3. Futuras tendencias del sector	11
3. Objetivos del proyecto	14
4. Metodología y recursos a emplear	15
5. La Cadena de Suministro	17
5.1. Definición.....	17
5.1.1. Objetivos de la cadena de suministro.....	18
5.1.2. Origen y antecedentes	19
5.2. Red de distribución	20
5.2.1. Definición	20
5.2.2. Factores que influyen en el diseño de una red de distribución	21
5.2.3. Estructuras y configuraciones de las redes de suministros	21
5.2.4. Omnicanalidad vs. Multicanalidad	25
5.2.5. Factores influyentes en la gestión de una red de distribución.....	27
6. Inteligencia Artificial	29
6.1. Definición.....	29
6.2. Machine Learning	30
6.3. Inteligencia Artificial en el sector retail.....	31
6.4. Data Analytics en el sector retail	34
7. Desarrollo del modelo de optimización	37
7.1. Introducción	37
7.2. Diseño de la red de distribución.....	37
7.3. Definición de la base de datos.....	40
7.4. Generador de pedidos.....	47

7.5. Generador de devoluciones	50
7.6. Generador de stock inicial.....	53
7.7. OMS	57
7.7.1. Escenarios y criterios de diseño	57
7.7.2. Escenario 1	60
7.7.3. Escenario 2	64
7.7.4. Escenario 3	69
7.7.5. Escenario 4.....	74
7.8. Análisis económico	79
8. Desarrollo del modelo de predicción	88
8.1. Introducción	88
8.2. Generación de datos	89
8.3. Modelos de predicción	100
8.3.1. Modelo ARIMA.....	102
8.3.2. Modelo FbProphet	108
8.4. Resultados	113
9. Conclusiones	115
10. Bibliografía.....	117
11. Anexos.....	119

Índice de Figuras

Figura 1. Ratio de penetración del sistema de venta online en los mercados globales.....	1
Figura 2. Países líderes en venta e-commerce del sector retail.....	2
Figura 3. Predicción de gastos en IT desde 2015 a 2020.....	3
Figura 4. Ventas de B2C e-commerce a nivel mundial.	5
Figura 5. Número de compradores online a nivel mundial.....	9
Figura 6. Selección del canal de compra en función del tipo de producto.	10
Figura 7. Cronograma de planificación del proyecto.....	16
Figura 8. Etapas de una cadena de suministro.	18
Figura 9. Antecedentes de la Cadena de Suministro [HABI08].	20
Figura 10. Almacenaje con el fabricante con envío directo.....	22
Figura 11. Almacenaje con el fabricante con envío directo y consolidación en tránsito.....	23
Figura 12. Almacenaje con el distribuidor con entrega por mensajería.	23
Figura 13. Almacenaje con el distribuidor con entrega a domicilio.	24
Figura 14. Almacenaje con el fabricante/distribuidor con recolección por parte del cliente. .	25
Figura 15. Localización de los centros de reposición primaria (Gibson & Defee, 2014).....	27
Figura 16. Inversiones en proyectos de Inteligencia Artificial. Fuente: McKinsey & Co., Accenture y CB Insights.	30
Figura 17. Principales beneficios de la IA en las empresas del sector. Fuente: Digiday UK..	33
Figura 18. Mapa de la red de distribución.	39
Figura 19. Representación de la integración de la BBDD.	46
Figura 20. Diagrama de flujo del generador de pedidos.....	49
Figura 21. Pedidos y devoluciones generadas para cada una de las tiendas.....	51
Figura 22. Diagrama de flujo del generador de devoluciones.	52
Figura 23. Exceso / Defecto de stock inicial por tienda.	54
Figura 24. Niveles de stock obtenidos con el generador de stock.	54
Figura 25. Diagrama de flujo del generador de stock inicial.....	56
Figura 26. Diagrama de flujo del OMS. Escenario 1.....	61
Figura 27. Diagrama de flujo del OMS. Escenario 2.....	66
Figura 28. Diagrama de flujo del OMS. Escenario 3.....	71
Figura 29. Diagrama de flujo del OMS. Escenario 4.....	76
Figura 30. Datos climáticos de la temporada 1.....	91
Figura 31. Datos climáticos de la temporada 2.....	92
Figura 32. Datos climáticos de la temporada 3.....	92
Figura 33. Calendario de eventos. Temporada 1.	93

Figura 34. Calendario de eventos. Temporada 2.	93
Figura 35. Calendario de eventos. Temporada 3.	94
Figura 36. Estudio de la relación entre el comportamiento del cliente y el clima.	95
Figura 37. Esquema de principios de la lógica del generador del histórico de pedidos.	98
Figura 38. Ingresos totales diarios para las tres temporadas simuladas.	99
Figura 39. Análisis de las unidades vendidas mensualmente en todo el histórico.	100
Figura 40. Diagrama de flujo del modelo general de aprendizaje supervisado. (Fuente: Sebastian Raschka, 2014).	102
Figura 41. ARIMA Model Results. ARIMA(5, 1, 1).	104
Figura 42. ARMA Model Results. Modelo ARIMA óptimo.	106
Figura 43. Predicción Modelo óptimo ARIMA(10, 0, 2).	107
Figura 44. Comparativa de datos de testeo y predicción. ARIMA(10, 0, 2).	107
Figura 45. Predicción de la demanda de unidades. Modelo FbProphet.	110
Figura 46. Componentes del modelo de predicción FbProphet.	112

Índice de Tablas

Tabla 1. Datos geográficos del almacén.	38
Tabla 2. Datos geográficos de las tiendas.	38
Tabla 3. Tabla de datos de tiendas. Parámetros retail.	41
Tabla 4. Extracto de la tabla de clientes.	42
Tabla 5. Familias de artículos de la BBDD.	43
Tabla 6. Subfamilias de artículos.	43
Tabla 7. Artículos.	44
Tabla 8. Extracto de la tabla de Pedidos.	48
Tabla 9. Extracto de la tabla de devoluciones.	50
Tabla 10. Extracto de la tabla de stock inicial.	55
Tabla 11. Resumen de los criterios de diseño de los escenarios del OMS.	59
Tabla 12. Distribución de pedidos origen/destino OMS. Escenario 1.	63
Tabla 13. Distribución de pedidos origen/destino OMS. Escenario 2.	68
Tabla 14. Distribución de pedidos origen/destino OMS. Escenario 3.	73
Tabla 15. Distribución de pedidos origen/destino OMS. Escenario 4.	78
Tabla 16. Costes de las actividades del OMS.	80
Tabla 17. Tiendas de la red de distribución. Modelo de predicción.	89
Tabla 18. Clientes. Modelo de predicción.	89
Tabla 19. Artículos de las tres temporadas de invierno. Modelo de predicción.	90
Tabla 20. Ventas mensuales por tienda. Modelo de predicción.	96
Tabla 21. Ventas diarias por tienda. Modelo de predicción.	97
Tabla 22. Histórico de pedidos. Modelo de predicción.	99
Tabla 23. Errores estadísticos modelo ARIMA(5, 1, 1).	104
Tabla 24. Análisis de sensibilidad. Modelo ARIMA.	105
Tabla 25. Análisis de sensibilidad. Modelo FbProphet.	109
Tabla 26. Análisis comparativo de los modelos de predicción.	113

Acrónimos y Términos

ARIMA – Autoregressive Integrated Moving Average

B2B – Business to Business

B2C – Business to Customer

CSM – Customer Service Management

DC – Data Center

DDSN - Demand-Driven Supply Networks

ERP – Enterprise Resource Planning

IA – Inteligencia Artificial

IoT – Internet of Things

MAE – Mean Absolute Error

MAPE – Mean Absolute Percentage Error

MSE – Mean Square Error

OMS – Order Management System

RFID – Radio-Frequency Identification

SCM – Supply Chain Management

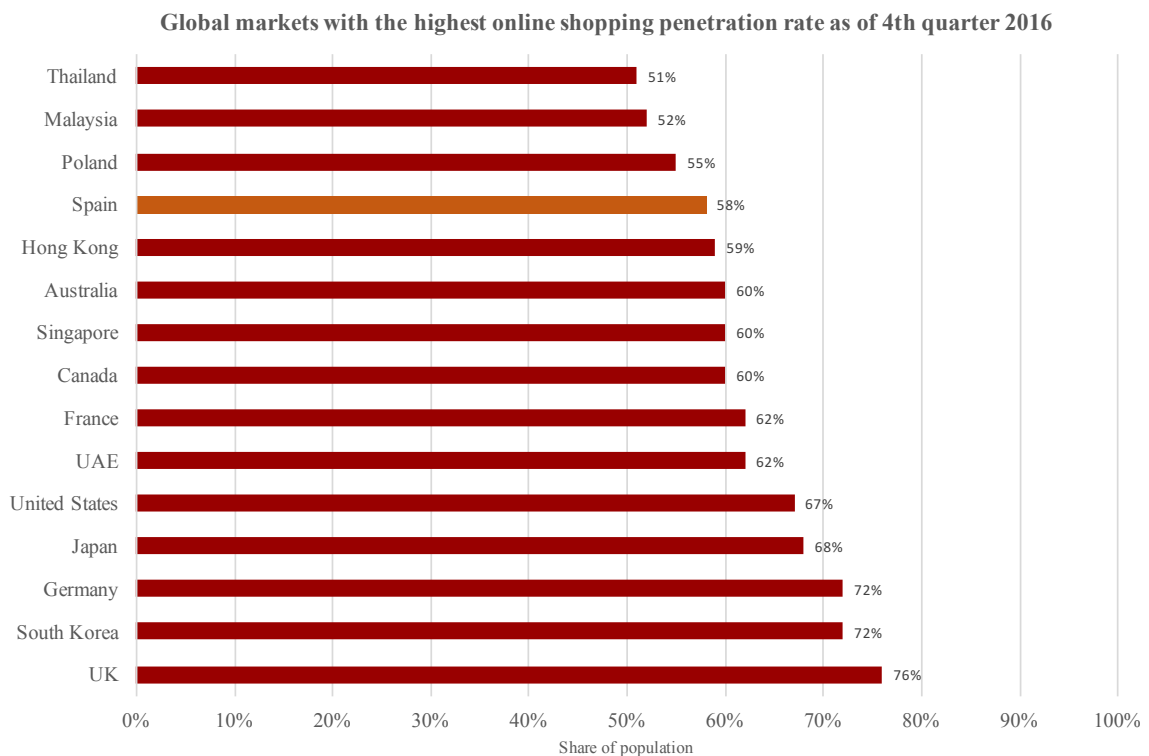
SKU – Stock Keeping Unit

1. Introducción

1.1. Contexto del sector

El sector de Retail ha experimentado en los últimos años un cambio considerable en su operación. El mercado rígido en el cual los consumidores debían de adaptarse a los productos que fabricantes y distribuidores ponían a su disposición, se encuentra totalmente obsoleto. Actualmente, el objeto fundamental de los fabricantes y distribuidores es satisfacer a sus clientes, pasando de una visión de “el producto en el centro” a “el cliente en el centro”. Como consecuencia, el comportamiento del mercado es cada vez más difícil de predecir y, por lo tanto, las empresas se ven forzadas a idear constantemente una nueva forma de satisfacer a los consumidores mediante estrategias competitivas.

Dentro de los múltiples factores que han motivado este cambio, uno de los más importantes es la globalización. Fruto del aumento continuo de la misma, existe una amplia oferta de valor que permite al consumidor elegir dentro de una extensa variedad de productos y servicios. El acceso por parte de las empresas a los mercados internacionales ha dado lugar a un mercado más competitivo, abierto y complejo. Como reflejo de este hecho sobre los mercados globales, se muestra en la Figura 1 el nivel de penetración de la venta online en dichos mercados [WILL15].



Source: Statista 2017

Figura 1. Ratio de penetración del sistema de venta online en los mercados globales.

En primer lugar, la posibilidad de diferenciarse ante los competidores es una tarea cada vez más difícil para las empresas ya que, actualmente, desarrollar una estrategia competitiva resulta un trabajo de elevado coste y riesgo. Con la fuerte inmersión del mercado chino, tanto a nivel tecnológico como económico competir con ellos requerirá establecer una serie de estrategias que permitan diferenciarse, ofreciendo una experiencia única en el proceso de compra. Un ejemplo del éxito del mercado chino en la venta e-commerce en el sector retail es el hecho ocurrido el día 11 de Noviembre de 2017, día en el Alibaba registró su record de ventas diario, cifrado en 25 billones de dólares [WANG17]. Además, en la Figura 2 se muestran los países líderes en niveles de ventas e-commerce en el sector retail. Debido a estos motivos, el objetivo actual de las empresas del sector es el de idear una táctica que les permita diferenciarse con respecto a la competencia, que frente a firmas como Amazon o Alibaba resulta una ardua tarea.

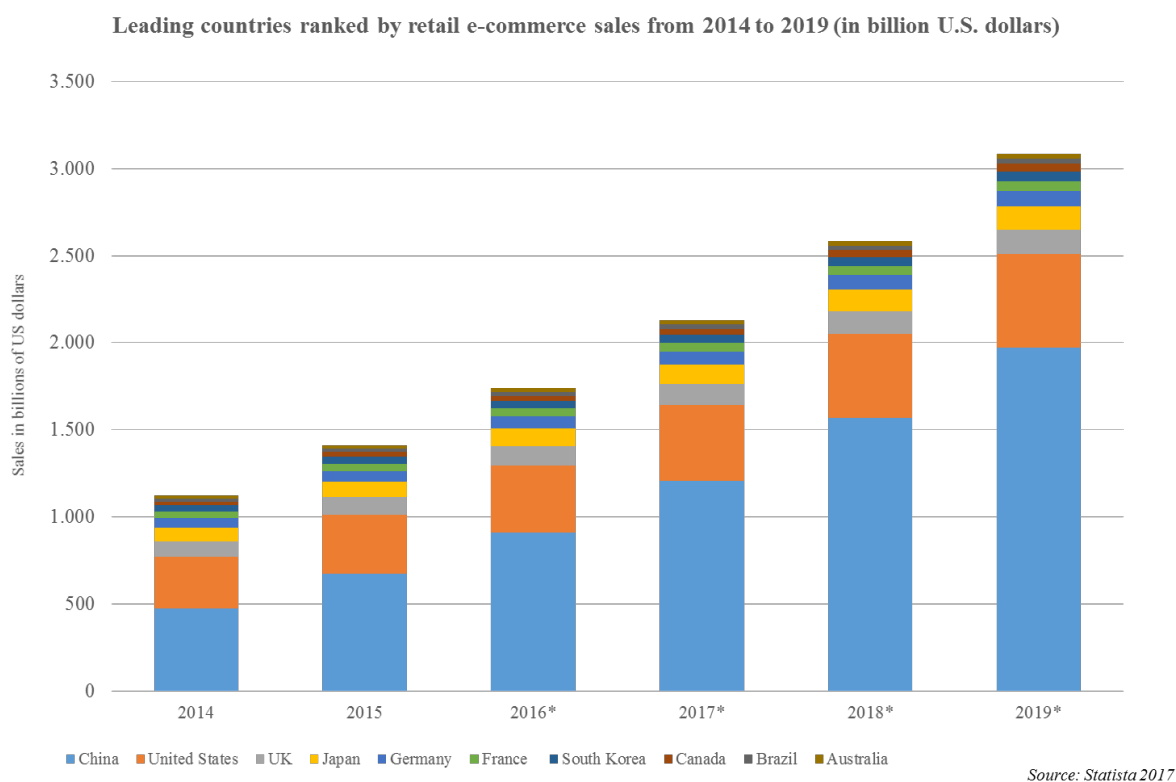


Figura 2. Países líderes en venta e-commerce del sector retail.

Otro aspecto característico del mercado actual es la visibilidad que existe dentro del mismo. Por un lado, el acceso a la información por parte del consumidor es plena, lo que les permite conocer y comparar la amplia gama de productos y servicios que las empresas ofertan. Cada vez es más habitual el utilizar las tiendas como escaparates para la venta online, el consumidor ve, toca, prueba los productos, recibe asesoramiento por parte de los vendedores y desde la misma tienda realiza la compra mediante su dispositivo móvil en el e-commerce, donde se le

ofrece mejores precios. Este hecho, junto con el cambio de mentalidad de los consumidores, da lugar a una reflexión previa a la compra. Tanto los factores sociales como propios del producto son relevantes a la hora de la selección. Por ejemplo, el impacto medioambiental a la hora de comprar un automóvil.

Finalmente, cabe señalar la complejidad que ha adquirido el mercado en relación con el estudio de su comportamiento. Actualmente, poder predecir cómo los parámetros que definen el mercado van a variar es realmente complejo, en esta línea, las empresas están apostando por el Big Data y la Inteligencia Artificial. A causa de los múltiples patrones de consumo existentes y de la gran variedad de factores que condicionan la compra, hacen que los especialistas tengan numerosos problemas para poder analizar las fluctuaciones existentes y ser capaces de definir una estrategia para poder cubrir la demanda de forma eficiente y balanceando faltas de stock (venta perdida) y sobrestock (inversión en inmovilizado y obsolescencia). Un ejemplo sería el cambio de comportamiento en los clientes con la variación del clima.

Otro hecho que ha sido causante del cambio de modelo de mercado es el avance de la tecnología en las últimas décadas, tal y como se muestra en la Figura 3. Con el desarrollo de los ordenadores, la aparición de los sensores, la automatización de procesos y las nuevas técnicas analíticas entre otros, el diseño de las cadena de suministro ha evolucionado significativamente frente al siglo pasado.

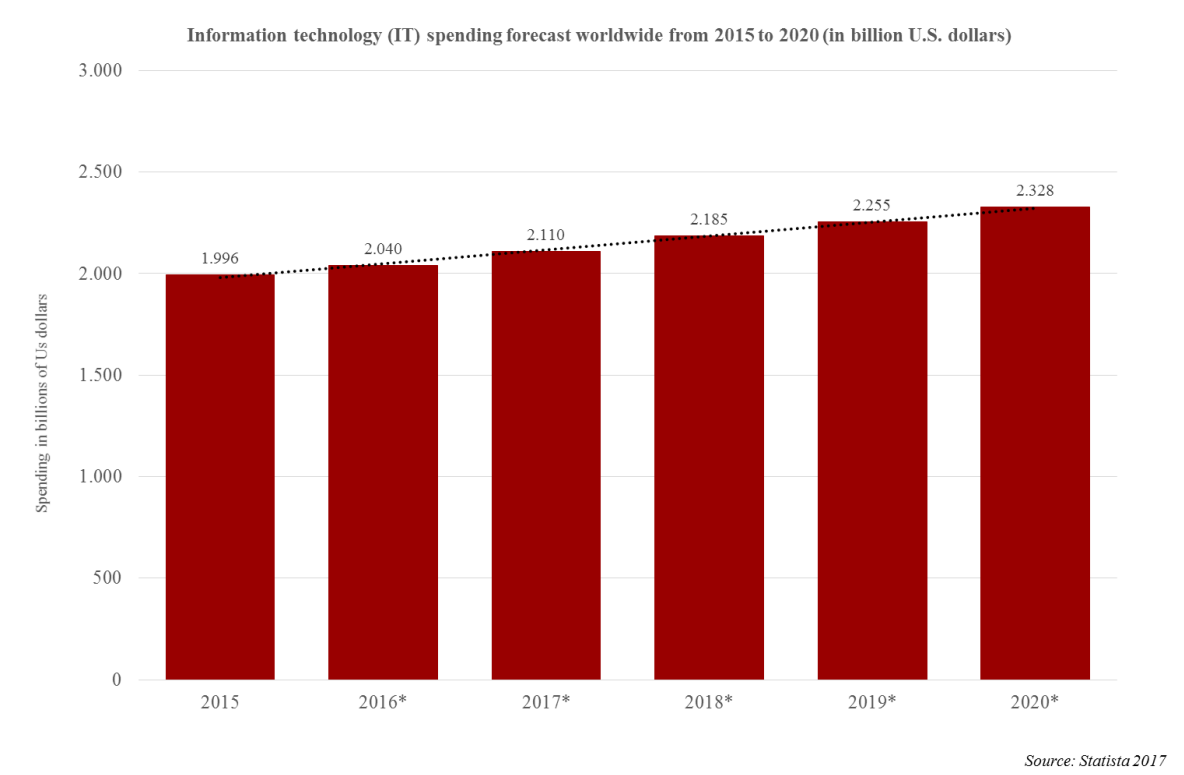


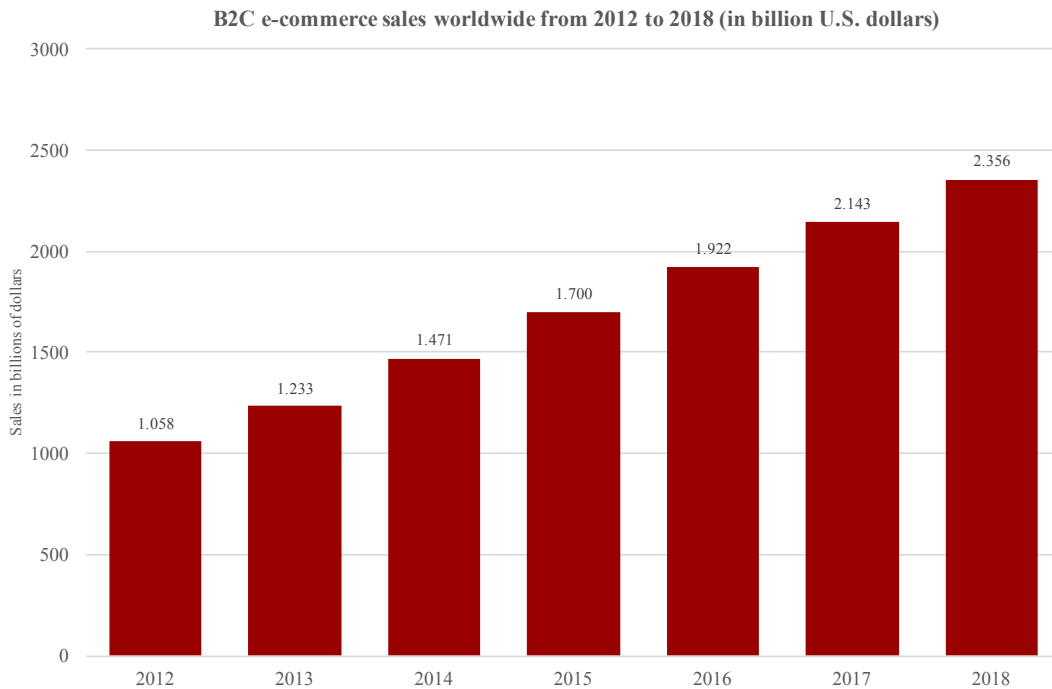
Figura 3. Predicción de gastos en IT desde 2015 a 2020.

A día de hoy es posible medir en tiempo real el comportamiento de cada uno de los componentes de la cadena, analizarlo y ser capaces de mejorar el proceso. Por tanto, los tiempos se reducen y, como consecuencia, los costes que cada uno de los procesos conllevan. Este hecho ha dado lugar a cadenas de suministro cada vez más complejas y que permiten a las empresas aumentar su producción y rendimiento, ahorrando costes.

Por otro lado, el flujo masivo de información entre los consumidores y las organizaciones ha provocado la aparición de nuevos medios de comunicación entre ambos y nuevos canales de comercialización por parte de las entidades de negocio. Además, se ha producido una revolución provocada por la aparición de nuevos e innovadores métodos de pago que permiten maximizar las oportunidades relacionadas con el flujo de dinero entre los usuarios y los proveedores. Todos estos aspectos han dado lugar a configuraciones casi ilimitadas de la cadena de suministro. Las compañías se ven obligadas a diseñar múltiples estructuras de cadena de suministro donde factores como la elección de un segmento concreto de consumidores es determinante en la elección de los canales de comunicación, método de pago, etc.

Teniendo en cuenta el panorama actual, se prevé que el crecimiento exponencial experimentado por las cadenas de suministro vaya a más. Resulta una ardua tarea para los profesionales estar en el estado del arte, no sólo en la parte técnica sino también desde el punto de vista de la gestión del cambio. Por tanto, las compañías deberán combinar su práctica estándar con la capacidad de tener flexibilidad ante urgencias específicas para evitar pérdidas económicas considerables o dar un mal servicio al cliente.

Las nuevas tendencias como el E-commerce, los drones para reparto, la tecnología “driverless” o “Internet of Things” serán las que condicionen el sector en los próximos años, actualmente algunas ya alcanzan niveles determinantes. Existe una estimación que prevé que el E-commerce representará pronto el 10% del total de las compras de Retail B2C en algunos países, que juntándolas con las que se llevan a cabo en B2B en la actualidad, el porcentaje estaría próximo al 20% [SERR17]. En la Figura 4 se muestra el crecimiento de las ventas online de B2C a nivel mundial en los últimos años.



Source: Statista 2017

Figura 4. Ventas de B2C e-commerce a nivel mundial.

1.2. Motivación del proyecto

El presente proyecto tiene por objeto principal llevar a cabo un proceso de optimización de la red de distribución, aplicable al sector Retail-Moda teniendo en cuenta toda la incertidumbre y variabilidad existente en el mercado actual.

La elección del tema responde a la necesidad existente de renovar y adaptar a las necesidades tecnológicas los sistemas de gestión de dicho sector debido a que se considera que esto puede provocar ahorros económicos considerables y un mejor servicio al cliente.

En la actualidad, las compañías pertenecientes al sector Retail llevan a cabo las actividades de distribución y de gestión de pedidos siguiendo una serie de prácticas que, en general, no están al nivel del desarrollo tecnológico y de las técnicas de optimización que a día de hoy existen. Sin embargo, el nivel de digitalización cada vez es mayor.

A nivel de distribución, existen una serie de factores que pueden ser condicionantes a la hora de la gestión de los pedidos y de la previsión de la demanda y que actualmente no se tienen en cuenta a causa de la dificultad que conlleva modelar esos aspectos con las herramientas que había hasta ahora. Por este motivo, la red logística de las organizaciones puede no estar totalmente optimizada y que su eficiencia sea menor de la que podría ser. Como consecuencia, esto da lugar a pérdidas económicas y a una mala gestión del inventario disponible, perdiendo oportunidades al no poner a disposición de los clientes el stock de toda la red de distribución.

Entendiendo el comportamiento del cliente y revisando la red de distribución que actualmente está implantada en las compañías, lo que se pretende es analizar con mayor detalle todos los factores que sean cuantificables y que puedan ser determinantes a la hora de gestionar los pedidos y de distribuir el stock a lo largo de los distintos canales y puntos de venta. Con esto, se podrán definir nuevas alternativas de optimización de la red logística que sigan patrones distintos a los empleados en la actualidad.

Como resultado, se pretende obtener un nuevo sistema de gestión pedidos que sea lo más eficiente posible. Esta solución se pretende llevar a cabo estableciendo una serie de indicadores para comparar y optimizar diversos esquemas de la red de forma precisa y apropiada que provoque un relevante impacto tras su implantación.

Por otro lado, se implementará un modelo que permite solventar un claro problema en la actualidad del sector, como es la predicción de la demanda. Se analizarán dos modelos basados en técnicas de Machine Learning, seleccionando uno de ellos tras previo análisis de resultados. Este modelo sería un complemento perfecto al modelo de optimización de la red de distribución descrito previamente.

2. Estado del arte

2.1. Metodología actual en redes de distribución

Tal y como se ha mencionado en el *Apartado 1.1: Contexto del sector retail*, el diseño de las cadenas de suministros ha adquirido un nivel de complejidad en los últimos años que hace que las configuraciones y estructuras que en la actualidad se emplean sean cada vez más diversas y difíciles de implementar con éxito. Como derivado de este hecho, las redes de distribución también están siendo diseñadas siguiendo el mismo patrón.

Un hecho relevante que propinó el cambio de las redes de suministro fue la aparición de los sistemas de venta multicanal. La aparición del e-commerce o el mobile commerce han propiciado cambios destacables en la gestión de las actividades de distribución debido a que suponen un aumento considerable en la complejidad del sistema. Modificaciones en la red logística o la construcción de nuevos almacenes de distribución son algunos de los cambios que las empresas del sector han tenido que afrontar.

Uno de los modelos más estandarizados es el uso de redes de abastecimiento dirigidas por la demanda (Demand-Driven Supply Networks, DDSN). Esta técnica permite entender el comportamiento del cliente y sus necesidades para poder reestructurar la red de abastecimiento. Para ello, es fundamental establecer un sistema de tecnologías y procesos que permite actuar en función de la demanda en tiempo real. Básicamente, se puede afirmar que el sistema de ejecución de las empresas del sector ha pasado de un modelo de empuje (*push*) a un modelo de extracción (*pull*) [ESTR07].

La gestión de la red de distribución de las empresas parte de la determinación de las necesidades de inventario y gestión del surtido con el fin de abastecer la demanda. Por ello, es cada vez más determinante tener la capacidad de predecir la demanda y anteponerse a las necesidades del cliente con el objetivo de proveerles una experiencia de compra adecuada. Históricamente, las empresas cubrían la incertidumbre de la demanda con inventario. Actualmente, esta táctica no sería viable debido a la magnitud de las empresas y debido al cambio radical del modelo de mercado. Con el establecimiento del sistema multicanal, la gestión del stock más comúnmente practicada es la de dividir el inventario para cada uno de los canales que la empresa disponga para facilitar su gestión. Sin embargo, esta estrategia en muchos casos resulta negativa ya que provoca problemas internos a nivel de entidad y un nivel de complejidad superior para la red de distribución [GIBS14].

En cuanto a la localización de los DC, la tendencia actual promovida por Amazon es la de situar cerca de cada una de las áreas metropolitanas importantes un DC. Con esta estrategia se consigue estar aún más cerca del cliente y, de esta forma, poder operar más con ellos.

Uno de los factores que determina la configuración y metodología seguida en la redes de distribución es el sector. Dependiendo del tipo de empresa, la configuración variará, a nivel

general, dentro de las estructuras de la red que se definirán en apartados posteriores del proyecto. Un ejemplo sería el uso de técnicas como *cross-docking* y *drop-shipping*, que consiguen acelerar el proceso de la cadena de suministros ya que no es necesario un almacenamiento intermedio. Con ello, las organizaciones son capaces de retrasar las necesidades de inventario hasta que la demanda real lo exija. Esta táctica suele ser empleada en grandes distribuidores de gran consumo como Carrefour o Grupo Eroski.

En la actualidad, las empresas emplean un sistema de software llamado OMS (*Order Management System*) que les permite seguir los pedidos desde el fabricante hasta su entrega al cliente final, gestionar el stock y mantener los niveles de inventario y los procesos de embalaje y envío. Además, permite sincronizar los pedidos a través de varios canales como, por ejemplo, un pedido que el cliente lleva a cabo a través del canal online y recoge en la tienda. Con la aplicación de este sistema se pretende que todas estas operaciones sean ejecutadas de forma eficiente y económica.

Uno de los programas de OMS más usado en la industria actualmente es el SAP Hybris Order Management, una solución desarrollada por SAP que permite a las empresas tener una visión en tiempo real de los niveles de stock en todos los canales implantados, mejorar la eficiencia a nivel de envíos y transporte y una serie de herramientas para el control y comunicación a lo largo de la red. También permite la integración con sistemas ERP, CRM y PIM debido a la arquitectura abierta que posee.

Un caso de éxito de implantación de esta solución es el de la empresa de ropa y calzado deportivo Li-ning, una de las compañías líderes en China. Con la implementación de los sistemas online en la empresa, la integración del sistema online con el offline en sus comienzos sufría una serie de carencias, hecho que se revertía en pérdidas económicas y en una mala gestión del stock. Para solucionar este hecho, la compañía optó por implementar la solución SAP Hybris B2C. Con esta implementación, la firma ha ganado en capacidades O2C (*Online-to-Offline*), creando un proceso de abastecimiento de los pedidos más eficiente. Como consecuencia, el volumen de ventas de Li-ning ha aumentado considerablemente y la rotación de inventario ha mejorado también. Además, su situación anterior se ha dado la vuelta, siendo ahora uno de los líderes en acometer una verdadera estrategia omnicanal.

2.2. Comportamiento del cliente

El cliente en el mercado actual juega un papel principal ya que se ha convertido en el centro de dicho mercado. Tal y como se ha comentado anteriormente, las empresas han cambiado su metodología de funcionamiento, pasando a un modelo totalmente focalizado en cubrir las necesidades de los consumidores. Existen actualmente tendencias sociales, tecnológicas, medioambientales, políticas y económicas que influyen tanto en cómo y cuándo los clientes llevan a cabo sus compras.

El avance de las tecnologías y el desarrollo de los nuevos canales de compra permiten a los clientes comprar cualquier producto o servicio en cualquier momento del día y desde cualquier dispositivo electrónico que tengan. Por otro lado, los criterios de compra de los consumidores cada vez son más personalizados, dadas las posibilidades que las marcas les ofrecen. Actualmente, a nivel de retail la mayoría de las empresas tienen servicios de personalización incorporados en sus procesos de venta. En la Figura 5 se muestra el crecimiento en los últimos años en el número de compradores a través del canal online.

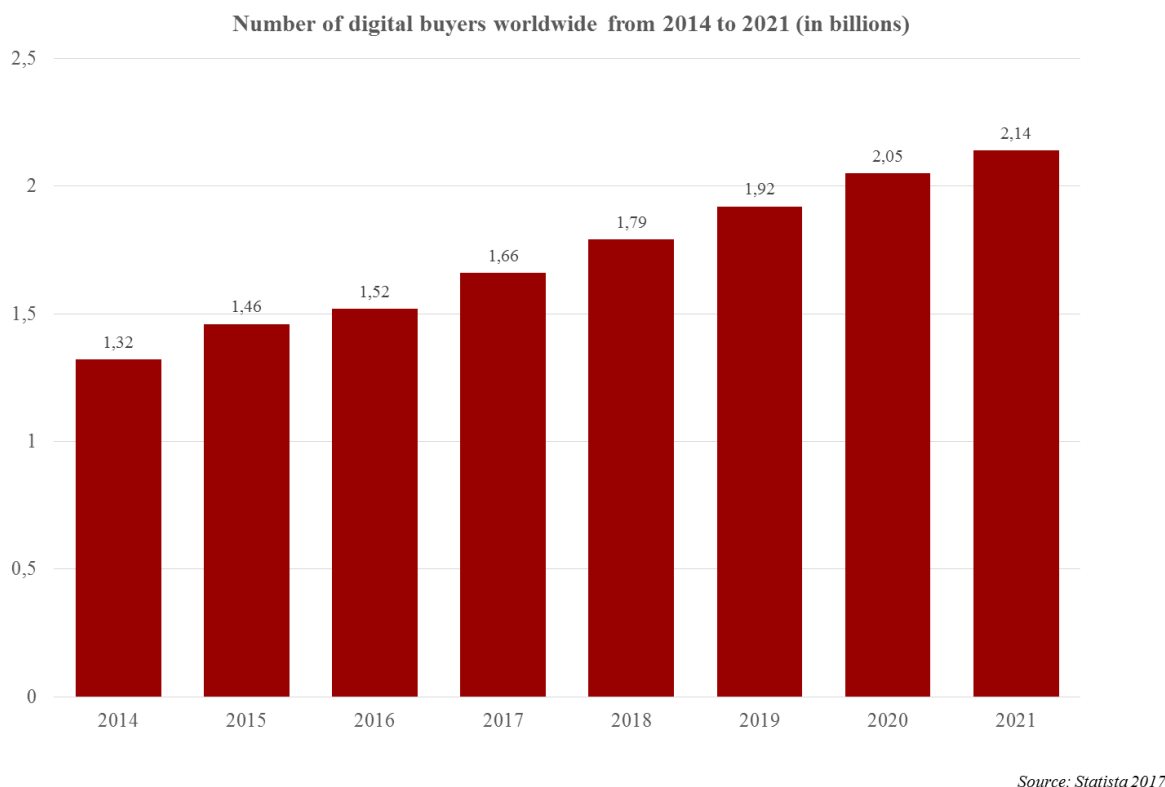


Figura 5. Número de compradores online a nivel mundial.

El uso que los consumidores hacen de los distintos canales que están disponibles a día de hoy depende del sector en el que el usuario se encuentre. En el sector de moda, los consumidores progresivamente están adoptando la idea de comprar a través de canales online debido a ciertos factores. En primer lugar, la facilidad con la que se pueden devolver los productos y el tiempo de devolución que las empresas ofrecen. Por otro lado, existen promociones sólo accesibles a través del e-commerce. Sin embargo, un gran porcentaje de los productos de este sector son actualmente adquiridos a través del canal físico debido a la rapidez y a la facilidad de comparar en el momento entre modelos y tallas.

En el sector alimenticio, la costumbre siempre ha sido la de comprar en tiendas físicas debido a que puedes seleccionar el producto al gusto del consumidor. Con la aparición de las plataformas de compra de supermercados como Tesco, en Reino Unido, y la reciente implantación de Amazon Eat, el cliente está tendiendo a usarlas para realizar compras de

grandes volúmenes en las cuales los gastos de envío son nulos o escasos en comparación con el importe total. Este tipo de plataformas de venta ofrecen a los clientes una experiencia de compra que les permite ajustar a su rutina esta actividad sin tener que perder tiempo en ella. Un caso singular relacionado con el comercio electrónico en el sector de la alimentación sería el de Mercado. A principios de Marzo de 2017, el máximo dirigente de la empresa, Juan Roig, declaró la intención de la empresa de potenciar su canal digital debido a las grandes carencias que considera que su compañía tiene en el aspecto tecnológico. El dirigente valenciano afirmó que “ *vemos que el comercio electrónico funciona y si logramos hacer que sea rentable iremos a por ello* ” [MERI17].

Por último, en otros sectores como la electrónica, la información que las empresas proporcionan a los consumidores es suficientemente detallada como para que ellos sean capaces de seleccionar el producto que estén buscando e incluso que puedan realizar comparaciones entre distintos modelos y marcas en función de las características que deseen. En este sector también se emplea la *best practice* comentada en apartados anteriores de usar las salas de venta como escaparates.

La tendencia que tienen los clientes en este sector es la de aumentar su participación en los canales online ya que esto les permite ajustar a sus rutinas y necesidades el proceso de compra. “Los clientes quieren vivir la misma experiencia cuando van a comprar a un mall que cuando compran de forma online” afirma Joe Bento, CSO de SEKO Logistics.

En la Figura 6 se lleva a cabo una comparación de los medios de compra en función de la industria, donde se refleja el comportamiento de los consumidores por tipo de producto.

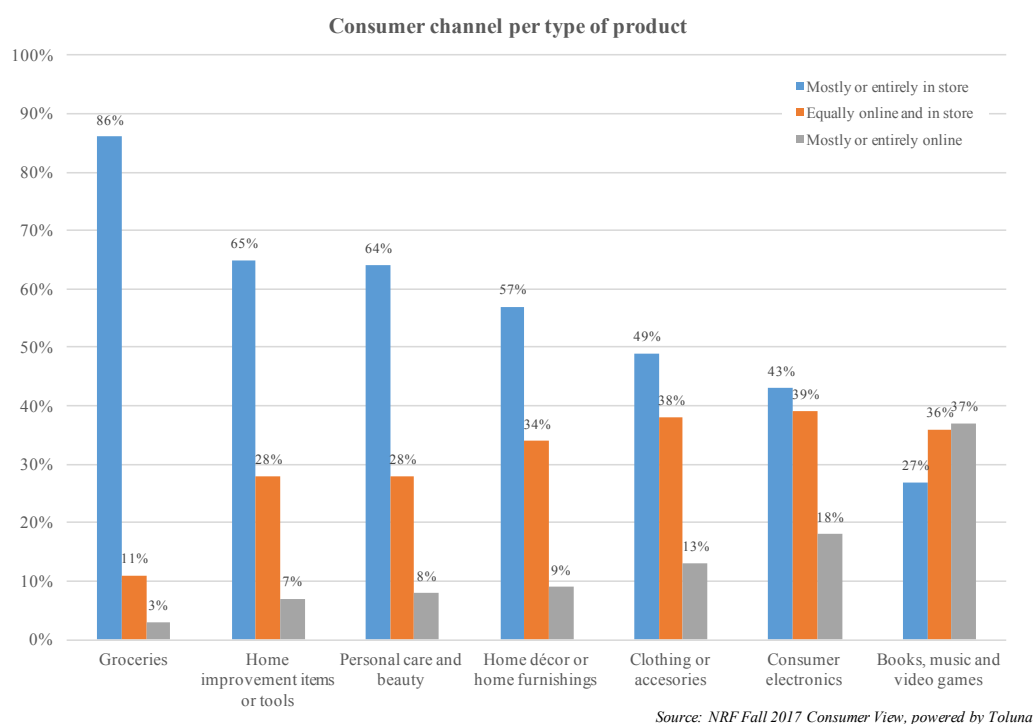


Figura 6. Selección del canal de compra en función del tipo de producto.

Por último, destacar el nivel de exigencia de los clientes, siendo este cada vez más acentuado en el aspecto de la velocidad de entrega de los pedidos. Esta tendencia afecta plenamente al sistema de entrega de última milla (“*last-mile delivery*”) y, como se comentaba anteriormente, en la localización de los DC en lugares cercanos a las áreas urbanas [VISS16]. Para la entrega de última milla están apareciendo nuevos métodos de entrega como el uso de nuevos medios de transporte, por ejemplo, las bicicletas, o el método *click-and-collect* (recogida en tienda de los productos pedidos a través de e-commerce o mobile commerce), que permiten llevar el pedido al cliente en menor tiempo, a menor coste y a la localización que más se adecue al cliente final. De cara al futuro, la implementación de soluciones innovadoras en esta área de logística será clave para diferenciarse respecto a la competencia ya que supone una parte importante del coste total del sistema de distribución [SIDE17].

Como consecuencia de todos los aspectos mencionados, el motivo fundamental de que existan tantos patrones de consumo y, por tanto, que el mercado sea tan complejo, es el comportamiento del cliente.

2.3. Futuras tendencias del sector

A causa de los continuos avances de los nuevos canales de venta que emplean las empresas del sector, e-commerce y mobile commerce, se habla de una revolución denominada Retail 3.0. Muchas de las empresas pertenecientes al sector están cambiando progresivamente su metodología con la que ofrecen sus productos o servicios al cliente. Con la aparición del concepto de la omnicanalidad, la experiencia que se le puede dar al cliente es incomparable ya que pueden llevar a cabo compras desde distintos canales de venta (e-commerce, mobile commerce, kioscos electrónicos, market places, etc.) compartiendo promociones, cestas de productos y muchas más ventajas.

El principio fundamental para la gestión de un sistema omnicanal es que el inventario sea común para todos los canales de venta. Por lo tanto, esto motivará a las empresas a reestructurar sus cadenas de suministros para adaptarse a este cambio tan relevante en su organización. Con la unificación del stock es posible conseguir un modelo robusto de negocio, tanto a nivel físico como del lado del e-commerce, afirman los expertos. Una referencia en la transformación de su modelo de negocio basado en la omnicanalidad es Walmart, que consiguió implementar soluciones tecnológicas en su red de distribución obteniendo resultados excepcionales [VISS16].

En cuanto a las opciones de transporte y distribución futuristas, éstas están siendo desarrolladas y probadas, pero se prevé que a corto plazo aún no se establezcan como una alternativa real debido a aspectos legislativos en su gran mayoría. Los coches autónomos y los drones se convertirán en los próximos años en la opción primaria para llevar a cabo las entregas de pedidos. Ejemplos de soluciones existentes es el caso de Cabify Express, servicio de mensajería

y entrega de paquetes urgentes [CABI17]. Esta estrategia sería claramente una forma de diversificación por parte de la empresa Cabify, aprovechando tu flota de coches para establecer la integración vertical en su compañía. Por otro lado, soluciones en desarrollo serían las de Amazon's "Primer Air" o Domino's Robotic Unit (DRU) [AMAZ16].

Todas estas propuestas de mejora y futuras estrategias de optimización tienen un objetivo común y es el de obtener una red de distribución dinámica. Para poder desarrollar este modelo es necesario monitorizar al detalle todos los procesos de la cadena de suministros con el fin de obtener una cantidad de datos suficientes para poder llevar a cabo análisis y estudios que permitan aumentar la flexibilidad y poder prever el comportamiento de sus clientes y del mercado en general.

Por otro lado, existe una posible tendencia con la instalación de sensores en el sector de la alimentación que permitan mantener un tratamiento adecuado del producto a lo largo de toda la cadena de suministros con el objetivo de que la entrega al cliente final se lleve a cabo en las mejores condiciones.

En relación con las redes de distribución, las compañías confían en el desarrollo de complejos softwares que permitan optimizar el inventario a lo largo de toda la red. Lo que se pretende es optimizar el flujo de inventario a la vez que se establecen modelos de predicción de demanda fiables que permitan combinar ambos aspectos y, como resultado, optimizar la red completa. Con esto se pretende dar solución a los problemas que actualmente existen en este sector a la hora de gestionar el SKU concreto en el lugar correcto, en la cantidad correcta y en el momento adecuado para que la demanda del consumidor sea cubierta correctamente.

Para poder llevar a cabo la construcción de estas herramientas es necesaria la implementación de las técnicas de IoT (*Internet of Things*) y de IA (Inteligencia Artificial). Con la implantación de estas soluciones se pretende:

- Construir estrategias más inteligentes de gestión cadena de suministros que combinen análisis de numerosa información relevante pertenecientes tanto al ámbito interno de la empresa como a factores externos.
- Reducir los costes operaciones y establecer unos niveles adecuados de disponibilidad del producto que permitan satisfacer la demanda del cliente.
- Grabar las transacciones de productos o servicios de alto valor mediante un sistema *block-chain* para proteger tanto al cliente como al vendedor.
- Obtener visibilidad del inventario en tiempo real que permita llevar a cabo procesos de optimización de predicción y reposición y gestión del inventario.

Herramientas como Siri, Microsoft Cortana o Amazon Alexa son usados en la actualidad por los consumidores para optimizar procesos de búsqueda de productos o servicios mediante

reconocimiento de voz. Los gigantes de la tecnología como Amazon, Google, Apple o Microsoft se encuentran actualmente en el proceso de desarrollo de potentes soluciones de IA pioneras en el sector.

En la Figura 2 se mostraban los niveles de ventas online en el sector retail de los países líderes en el sector. Estos datos pueden ser un perfecto indicador de la transformación que está sufriendo la industria debido a la implantación y elevado desarrollo de las tecnologías, sobre todo en cuanto a los canales de venta se refiere.

3. Objetivos del proyecto

Los objetivos que se pretenden cumplir con la realización de este proyecto se indican a continuación:

- Analizar las metodologías actuales que se emplean en las redes de distribución con el objetivo de identificar debilidades y aspectos que mejorar.
- Desarrollar una prueba de concepto que permita identificar los aspectos generales que deben ser tenidos en cuenta a la hora de la construcción de la solución final.
- Diseño de una solución que permita la optimización de las redes de suministros y la gestión del stock y surtido de las empresas pertenecientes al sector retail.
- Implementar herramientas de análisis de Machine Learning al modelo para estudiar los datos generados e identificar posibles patrones del comportamiento de la red de distribución.
- Determinar el impacto económico de la implantación de la solución propuesta. Se presentarán distintas situaciones caracterizadas por diversos parámetros que deberán ser analizadas con el fin de identificar el óptimo.
- Desarrollar un modelo de predicción que permita, partiendo de un histórico de datos, anticiparse a las ventas en temporadas próximas teniendo en cuenta factores como el clima o las promociones.
- Analizar y validar distintos modelos de predicción basados en técnicas de Inteligencia Artificial en función de los objetivos buscados y los resultados obtenidos.

4. Metodología y recursos a emplear

En primer lugar, será necesario definir una red de distribución sobre la que se van a llevar a cabo los estudios correspondientes del proyecto. La definición de la misma implica establecer los centros de distribución y las tiendas que la van a conformar, determinando su localización geográfica, sus niveles de ventas totales y otros parámetros como por ejemplo el ticket medio o las unidades por ticket (UPT). Por otro lado, habrá que definir los artículos del inventario y los clientes, distinguiendo entre aquellos que compran en tienda física o los que lo hacen a través del canal online.

Una vez definida la red, se realizará una prueba de concepto en la cual se construirá un generador aleatorio de pedidos, devoluciones y stock inicial tomando como datos de partida los definidos en los maestros. Con todo esto lo que se pretende es definir el punto de partida de nuestra red de distribución. Además del generador de pedidos, se construirá una prueba del modelo que se pretende diseñar en la cual se plantearán diferentes escenarios relacionados con la gestión de la red de distribución, que estarán parametrizados de distinta forma con el fin de contabilizar distintos factores que pueden ser optimizados a la hora de obtener una gestión eficiente. Esta prueba se desarrollará haciendo uso del lenguaje de programación Python. Por lo tanto, se diseñará un código que permita, por un lado, generar la situación inicial de la red de distribución y, luego, pueda optimizar en base a los datos generados la gestión de la misma con el objetivo de abastecer la demanda de forma eficiente.

Una vez definida la prueba de concepto, se pretende definir un modelo tomando como base dicha prueba y en el que se implementarán herramientas de Machine Learning con el objetivo de optimizar y analizar la red de distribución. Con esto lo que se pretende es darle una solidez al modelo hasta tal punto que pueda ser capaz de optimizar de forma automática con herramientas de predicción y análisis pertenecientes al campo de la Inteligencia Artificial.

Por último, se pretende evaluar económicamente cada uno de los escenarios con el fin de determinar qué factores son los óptimos a tener en cuenta para gestionar el sistema.

Cabe destacar que los datos que conforman los maestros del sistema que se definirán en el modelo, la red de distribución, los pedidos y los modelos de determinación de la fuente de suministros serán definidos y contruidos en base a datos y técnicas reales del mercado. Con esto se pretende que el modelo sea lo más cercano a la realidad y que su comportamiento exprese rasgos y funciones que podrían ser aplicadas en cualquier empresa del sector.

Se adjunta en la siguiente página un cronograma de la programación del proyecto de acuerdo con lo definido en las reuniones con el director.

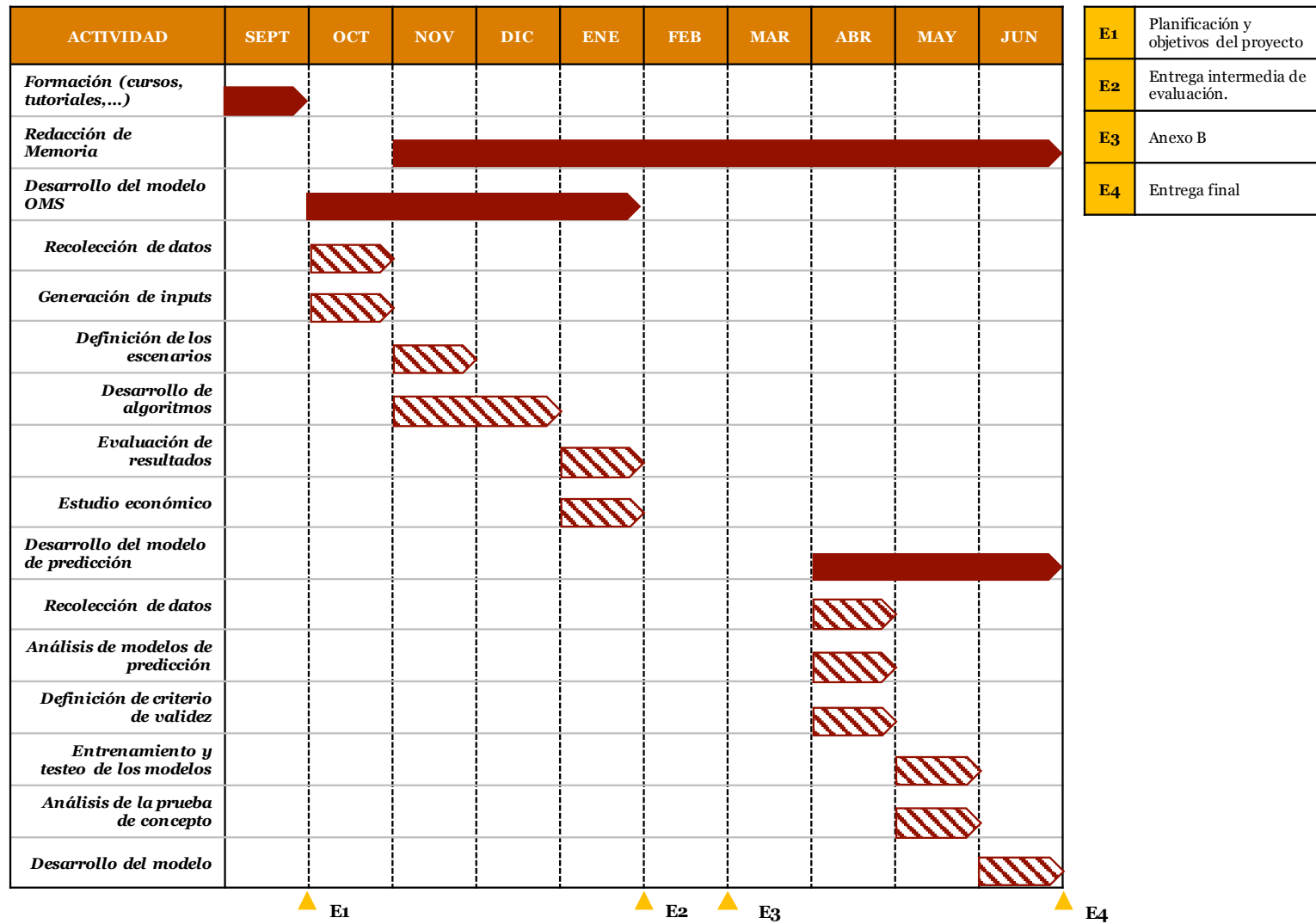


Figura 7. Cronograma de planificación del proyecto.

5. La Cadena de Suministro

5.1. Definición

La cadena de suministro es una red dinámica que implica un flujo continuo entre individuos, organizaciones, recursos, actividades y tecnologías involucradas en la fabricación y venta de un producto o servicio. El proceso completo está definido desde la entrega de la materia prima por parte de un proveedor al fabricante hasta la entrega del producto terminado o servicio al consumidor final. La cadena de suministro no sólo incluye los proveedores y los fabricantes, sino que también está formada por los transportistas, almacenes, minoristas e, incluso, los propios clientes.

El objetivo fundamental de la cadena de suministro es satisfacer las necesidades del cliente. Por este motivo son una parte integral de la misma. El término cadena de suministros no sólo hace referencia al flujo de un producto o de un servicio desde un proveedor al cliente pasando por el fabricante, distribuidor y minorista, sino que también incluye el flujo de información, productos y dinero. Además, en cada una de las etapas de la misma no hay una única entidad sino que hay una gran variedad de proveedores y distribuidores.

Tal y como se muestra en la Figura 8, la estructura típica de una cadena de suministro está formada por varias etapas que están interrelacionadas entre ellas formando una red estructural. Las etapas que incluye son las siguientes:

- Clientes.
- Minoristas.
- Almacenes y distribuidores.
- Fabricantes.
- Proveedores.

El flujo entre cada una de las etapas que conforman la cadena puede ir en ambas direcciones en ciertas ocasiones.

No todas las etapas descritas en la Figura 8 tienen que estar presentes en la cadena suministro, todo dependerá de las necesidades del cliente y del rol que tenga cada una de las etapas involucradas. Por ejemplo, en el caso de las empresas del sector retail, será necesaria la incorporación de almacenes dentro de la estructura de su red de suministro.

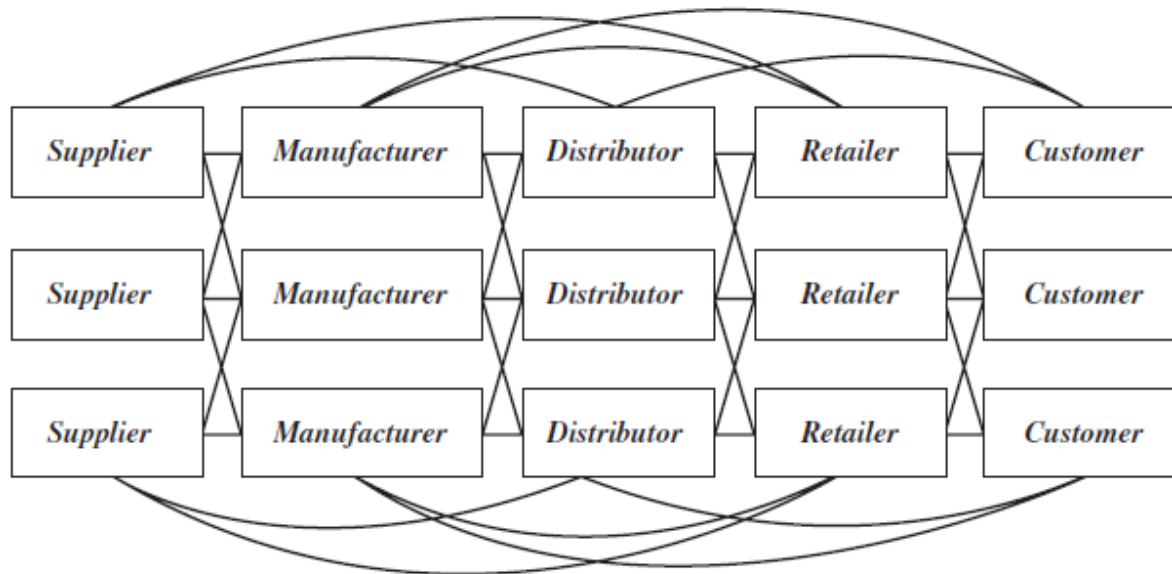


Figura 8. Etapas de una cadena de suministro.

El objetivo fundamental de toda cadena de suministro es el de maximizar beneficios, que sería la diferencia entre el valor que tiene el producto o servicio final para el cliente y el coste que ha supuesto cumplir con los requerimientos establecidos por el cliente.

El alcance de los objetivos en una cadena de suministro se fundamenta en una buena gestión de la misma (conocida en inglés como SCM). Se define la gestión de la cadena de suministro como la racionalización activa de las actividades del lado de la oferta de un negocio para maximizar el valor del cliente y obtener una ventaja competitiva en el mercado. Esto se consigue gestionando eficientemente los bienes y productos de la cadena y los flujos de información y fondos ya que cada uno de ellos genera costes dentro de la misma.

Normalmente, SCM intenta controlar o vincular centralmente la producción, el envío y la distribución de un producto. Una buena gestión de la misma permite reducir costos excesivos y entregar los productos al consumidor más rápido. Para ello es necesario mantener un control estricto de los inventarios, la producción interna, distribución y ventas. Actualmente, las empresas consideran un valor añadido a sus operaciones una buena gestión de la cadena de suministro.

5.1.1. Objetivos de la cadena de suministro

Los objetivos más destacables de la cadena son los siguientes:

- Disminuir los costes de transporte, distribución y stock de materias primas y productos tanto semiterminados como terminados. Una cadena de suministros eficiente conoce los intercambios, sinergias y costes producidos a lo largo de ella.

- Alcanzar el máximo rendimiento en cada una de las actividades que la componen con el fin de que los flujos sean rápidos, los stocks mínimos, que permitan disminuir tanto el lead time como el takt time y que permita mejorar las relaciones tanto con el cliente como los proveedores a través de la prestación de productos y servicios de calidad.
- Compenetrar las decisiones estratégicas (largo plazo) con las tácticas y operativas (medio y corto plazo) de la forma más eficiente.
- Maximizar el intercambio de información y conocimiento.

5.1.2. Origen y antecedentes

El término “cadena de suministro” (del inglés: Supply Chain) apareció en la década de los 80s cuando el consultor Keith Oliver lo empleó en una entrevista para Financial Times. En 1985, Houlihan lo definió como un sistema de entidades (proveedores, fabricantes, almacenadores, distribuidores, vendedores y clientes) en los cuáles existe un flujo de materiales y la información fluye en ambas direcciones. Durante el siglo 20, este concepto tuvo especial importancia con la creación de la línea de montaje.

Existen hechos que pueden ser considerados como antecedentes, tal y como se muestran en la Figura 9. En las décadas de los 50s y 60s en las cuales se desarrolla la estrategia de la producción en masa con el fin de minimizar los costes unitarios de producción. En esta época, la idea de cooperar y colaborar con proveedores y clientes no se consideraba.

En el siglo siguiente, con la aparición de las primeras computadoras comenzaron a aparecer ideas tecnológicas y herramientas matemáticas de optimización relacionados con los sistemas de distribución.

En la década de 1990, surgieron los sistemas de gestión de recursos humanos y materiales, conocidos como ERPs. Además, con la aparición del Internet, se desarrollaron los sistemas de colaboración e integración con lo que se produjo un aumento del valor añadido.

En los últimos años con la globalización han aparecido múltiples estrategias de gestión de las cadenas de suministros con objeto de controlar y definir complejos sistemas de distribución.

La aparición de este concepto ha permitido aumentar la competitividad de las empresas en los siglos XX y XXI, siendo actualmente una ventaja competitiva claramente definida en las empresas que tienen una fuerte componente de distribución.

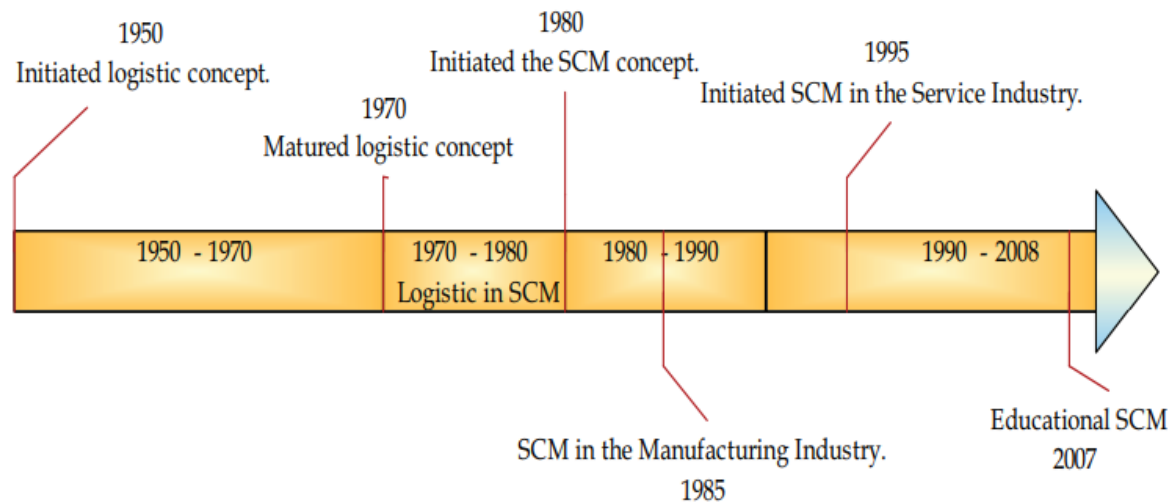


Figura 9. Antecedentes de la Cadena de Suministro [HABI08].

5.2. Red de distribución

5.2.1. Definición

La red de distribución comprende todos los pasos que sigue el producto desde que es recibido del proveedor hasta que se pone a disposición del cliente. El diseño de la misma estará influenciada por diversos factores, como pueden ser la disponibilidad del producto, el tiempo de respuesta, la variedad de los productos, etc. La distribución es clave en la rentabilidad general de una firma ya que afecta tanto al coste de la cadena de suministro como a la experiencia del cliente. Según Sunil Chopra y Peter Meindl, la distribución se corresponde aproximadamente con el 35% de las ganancias de una empresa del sector retail. Un claro ejemplo de ello es Wal-Mart, una de las compañías con mayor rentabilidad atribuye su éxito a su excelente diseño de la red de distribución y operaciones. [CHOP13]

La cadena de suministros establece relaciones entre muchos componentes que, en muchos casos, parecen no tener nada que ver entre ellos, pero guardan una relación importante. Principalmente, se consideran los siguientes:

- Inventarios.
- Transporte.
- Infraestructura.
- Información.

Uno de los principales objetivos para obtener una cadena de suministros eficiente en cuanto a costes y tiempo es establecer el equilibrio entre cada uno de sus componentes.

5.2.2. Factores que influyen en el diseño de una red de distribución

A alto nivel, para evaluar el funcionamiento de una red de distribución deberían de tenerse en cuenta dos dimensiones:

- Las necesidades que tiene el cliente.
- El coste que implica cubrir las necesidades que tiene el cliente.

Por otra parte, la satisfacción del cliente está influida por muchos factores que están directamente relacionados con la estructura de la red de distribución y que son determinantes a la hora de crear una nueva red. [TOMP09]

- El tiempo de respuesta de una orden. Es el tiempo que tarda el cliente en recibir un pedido.
- Variedad de productos. Número de productos o configuraciones ofrecidas por la red de distribución.
- Disponibilidad de los productos. Probabilidad de tener un producto en el inventario cuando la orden del cliente llega.
- Experiencia del cliente. Facilidades que tiene el cliente en el proceso de compra y recepción de pedidos.
- Time to market. Tiempo que tarda un producto nuevo en entrar en el mercado.
- Visibilidad de los pedidos. Posibilidad de seguir el pedido a lo largo de todo el proceso de entrega.
- Retornabilidad. Afecta tanto al cliente como al sistema de la empresa en relación con la tramitación de la devolución.

5.2.3. Estructuras y configuraciones de las redes de suministros

Según la industria existen diversos tipos de configuraciones de la red de distribución. A continuación, se muestra una explicación de cada una de ellas y su esquema principal. [CHOP13]

1. Almacenaje con el fabricante con envío directo.

Esta opción permite el envío directo del fabricante al consumidor final. El fabricante es el que mantiene inventarios mientras que el minorista es el que gestiona la información que circula dentro de la red de distribución. Por este motivo, es necesario una buena infraestructura de información.

La gran ventaja es la capacidad de centralizar los inventarios ya que permite proporcionar un alto nivel de disponibilidad con bajos niveles de inventario. Esta configuración es beneficiosa

para los productos con baja demanda y alto valor. Por otro lado, permite al fabricante posponer la personalización hasta que se coloca el pedido. Esta estrategia la lleva muy bien a cabo Dell. Por el contrario, los costos de transporte son altos debido a que las distancias son grandes. Como consecuencia de este hecho, los tiempos de respuesta también serán mayores.

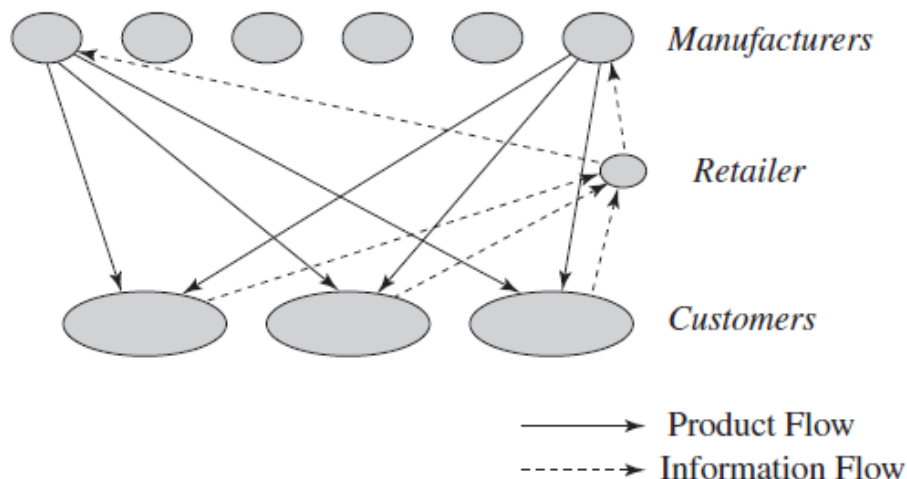


Figura 10. Almacenaje con el fabricante con envío directo.

2. Almacenaje con el fabricante con envío directo y consolidación en tránsito.

A diferencia del envío directo, la consolidación en tránsito combina los productos provenientes de distintas localizaciones de fabricantes, de forma que sólo se lleva a cabo una entrega. La información se gestiona siguiendo la misma metodología, con lo que la infraestructura debe ser muy importante y compleja para que las operaciones entre fabricantes, minorista y transportista sean coordinadas.

Con esta configuración también se agregan inventarios en el fabricante, lo que permite posponer la personalización. Este método tiene grandes beneficios para los productos de alto valor y cuya demanda es difícil de predecir. Además, la experiencia del cliente probablemente sea mayor.

A diferencia de la estrategia anterior, el coste de instalación es un poco más alto al igual que el tiempo de respuesta ya que es necesario consolidar el pedido antes de enviarlo.

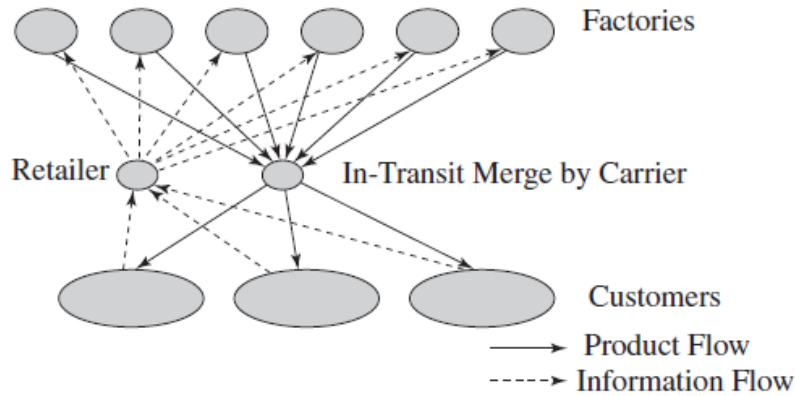


Figura 11. Almacenaje con el fabricante con envío directo y consolidación en tránsito.

3. Almacenaje con el distribuidor con entrega por mensajería.

En esta configuración, el inventario se encuentra en almacenes intermedios de distribuidores/minoristas y los productos se transportan al cliente final a través de servicios de mensajería. Por ejemplo, Amazon emplea este método en combinación con el envío directo desde el fabricante. Este sistema requiere niveles de inventario más altos ya que, en general, agrega incertidumbre a la demanda. Suele emplearse cuando se trata con artículos de rotación media a rápida.

A diferencia de los anteriores, los costes de transporte son algo más bajos debido a que pueden emplearse medios más económicos y, por otro lado, es factible combinar en un solo embarque los pedidos que van al cliente final. Además, la infraestructura de información que se requiere es menos complejo, lo que implica menores costes. Como última ventaja, el tiempo de respuesta es menor ya que se adquiere mayor cercanía al cliente final con esta configuración. Sin embargo, los costes de instalaciones y procesamiento son algo superiores ya que se incorporan almacenes intermedios.

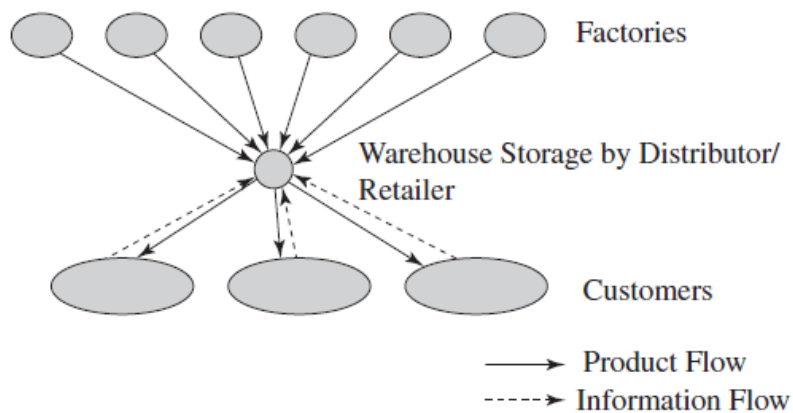


Figura 12. Almacenaje con el distribuidor con entrega por mensajería.

4. Almacenaje con el distribuidor con entrega a domicilio.

Este tipo de entrega se lleva a cabo por parte del distribuidor/minorista en el hogar del cliente. Por este motivo, la experiencia de los mismos es muy buena. Se suele emplear cuando se tienen artículos de alta rotación, como por ejemplo los pertenecientes a la industria de los comestibles, donde la variedad de los productos suele ser, en general, más baja. Los niveles de inventario son muy altos ya que la desagregación es mucho menor.

A diferencia de la entrega con transportista, esta tipología requiere que la cercanía al cliente sea mucho mayor, por lo que el número de almacenes debe de ser aún mayor. Esto provoca que el coste de instalaciones y procesamiento también aumente considerablemente pero que el tiempo de respuesta sea más rápido. Por otro lado, la infraestructura de información es similar a la del caso anterior. Sin embargo, requiere capacidades adicionales de programación de entregas.

Por último, destacar que para esta configuración la retornabilidad es la mejor ya que se puede recoger las devoluciones a la vez que se entrega un producto.

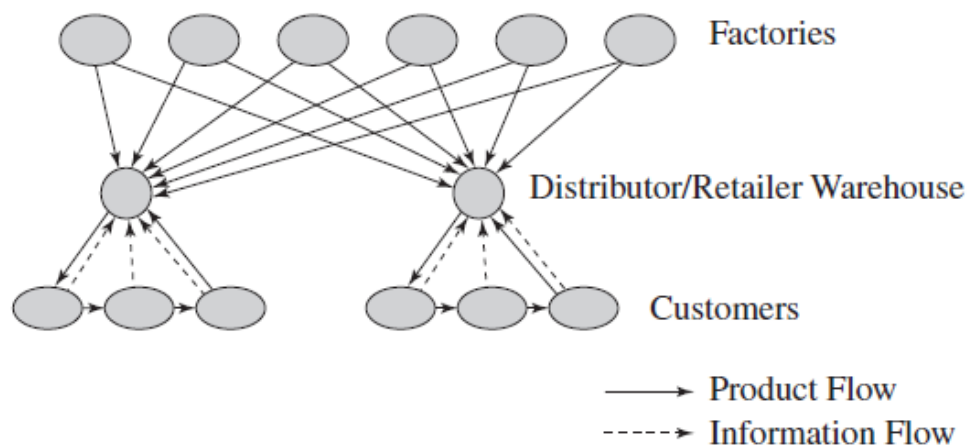


Figura 13. Almacenaje con el distribuidor con entrega a domicilio.

5. Almacenaje con el fabricante/distribuidor con recolección por parte del cliente.

Esta configuración permite a los clientes recoger los pedidos que realizan en un punto de surtido designado por la empresa correspondiente. Esto hace que la experiencia del cliente sea algo más baja ya que tiene que recoger su pedido en el punto correspondiente, pero tiene mayor flexibilidad que si se lo enviasen a domicilio.

Los costes de inventario pueden mantenerse bajos ya que es posible tener los inventarios de alta rotación almacenados en los puntos de surtido, mientras que los de baja rotación se encuentran en los almacenes de distribuidores o en el fabricante. Por otro lado, como se concentran los productos en el punto de surtido, la agregación es mayor y, con ello, el coste de transporte será menor.

En cuanto a la infraestructura de información, será necesaria una buena coordinación entre cada una de las partes de la red de distribución. Además, la visibilidad en esta configuración es bastante importante ya que es necesario tener al cliente informado. Dicho sistema puede resultar complejo de implementar ya que requiere la integración de varias etapas de la cadena de suministro.

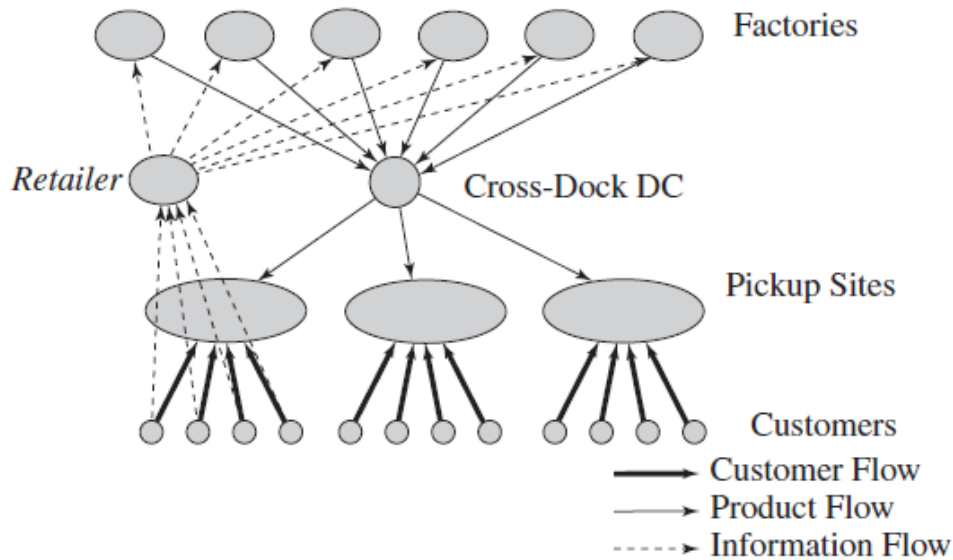


Figura 14. Almacenaje con el fabricante/distribuidor con recolección por parte del cliente.

6. Almacenaje con el vendedor con recolección por parte del cliente.

Es considerada como la cadena de suministro más tradicional. El inventario se almacena en las tiendas, donde los clientes pueden comprar directamente o hacer un pedido para luego recogerlo allí. Esta configuración es más apropiada para artículos de alta rotación o para aquellos que el cliente valora una respuesta rápida.

Tiene como ventaja que el coste del transporte es mucho menos debido al empleo de medios de transporte más baratos y, también, que el almacenamiento es local. Como consecuencia, los tiempos de respuesta son más bajos.

En contraposición, el almacenaje local incrementa los costes de inventario ya que existe mayor desagregación. Resulta más caro, en comparación con las demás configuraciones, mantener un alto nivel de disponibilidad del inventario. Además, las infraestructuras tanto físicas como de información son mayores y, por lo tanto, sus costes también lo son.

5.2.4. Omnicanalidad vs. Multicanalidad

Con el desarrollo tecnológico de las últimas décadas, los comercios han experimentado un cambio importante en relación con sus canales de venta y de acercamiento al cliente. Con la

aparición de Internet, los teléfonos móviles y el resto de dispositivos electrónicos y tecnologías, las empresas del sector retail cuentan con una variedad de canales de venta bastante amplia a parte del físico.

El objetivo de la estrategia multicanal es aprovechar las características de cada medio y explotarlas para mostrar el producto o servicio que se quiere mostrar en distintos formatos. En una solución de este tipo, cada uno de los medios tienen sus correspondientes clientes y están estructurados y organizados de forma individual. Por ejemplo, el stock de la venta offline es distinto al de la venta por Internet.

A diferencia de la multicanalidad, la omnicanalidad tiene como objetivo crear una mejor experiencia del cliente. Se trata de emplear tácticas que aprovechen mejor los canales que tiene el negocio para proveer una experiencia más consistente fundamentada en los siguientes aspectos:

- Establecer continuidad entre las plataformas de venta por móvil o por ordenador. Es decir, que puedas mantener los carritos de compra entre ambas plataformas o los últimos artículos vistos.
- Misma promoción en todos los canales. Aumenta la probabilidad de que el cliente vea la oferta y, por tanto, pueda comprar el producto o servicio anunciado.
- Permitir pedidos online con recogida en tienda.
- Interfaces comunes entre los distintos canales de venta.

Este modelo permite a las empresas ajustar más fácilmente las órdenes de pedidos directas de los clientes a las rutinas de selección de pedidos existentes en el DC (“Data Center”). A la misma vez, aporta a aquellas entidades con gran porcentaje de reabastecimiento mayor agilidad y facilidad para combinar la reposición y los pedidos dentro de la misma operación.

En la Figura 15 se muestran los resultados de un estudio llevado a cabo por Auburn University y RILA (Retail Industry Leaders Association), en el cual se presentan los distintos métodos que emplean las empresas del sector retail para completar las órdenes de pedidos de los clientes. Se observa que gran parte de los pedidos se satisfacen desde los DC, favoreciendo la integración horizontal en los sistemas de distribución. [GIBS14]

Una de las barreras fundamentales para los negocios al establecer el cambio hacia un modelo omnicanal es su elevado coste y complejidad. Su implantación implica un cambio significativo tanto a nivel tecnológico como a nivel de organización empresarial.

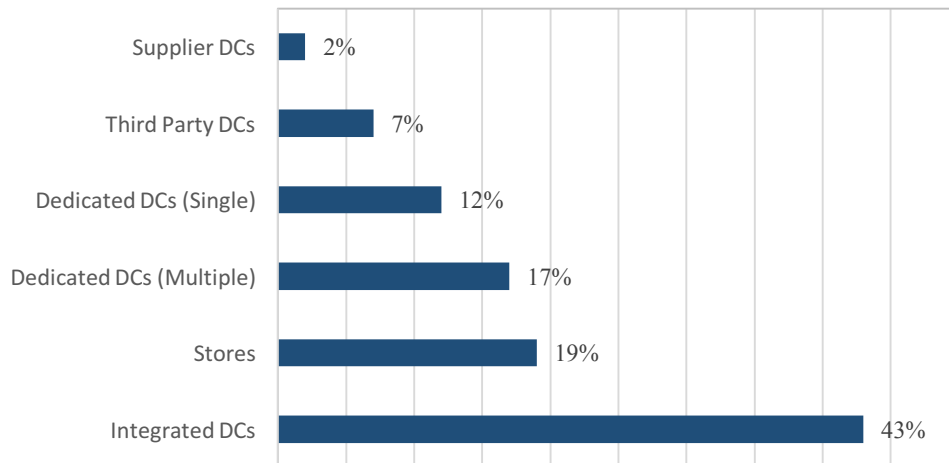


Figura 15. Localización de los centros de reposición primaria (Gibson & Defee, 2014).

5.2.5. Factores influyentes en la gestión de una red de distribución

La gestión del inventario a lo largo de la red de distribución de una compañía depende fundamentalmente de la demanda. La práctica habitual en las empresas del sector se basa en distribuir el stock entre sus tiendas en función de los patrones que recopilan en sus históricos. Con esto lo que pretenden es que el nivel de inventario sea el necesario para cubrir la demanda sin que se produzcan acumulaciones de stock. Sin embargo, para llevar a cabo una gestión eficiente de la red de distribución se han de tener presente una serie de factores que permitirán que el rendimiento de la compañía mejore y, como consecuencia, que los resultados financieros también lo hagan.

A nivel de reposición de tiendas, los principales aspectos que se tienen en cuenta son la cercanía del almacén a la tienda y, además, que en el almacén haya disponibilidad del artículo solicitado en stock. En caso contrario, será necesario iterar por cada uno de los centros de distribución de la red con el fin de optimizar los costes de transporte que, en este caso, serán los más significativos.

Otro aspecto a considerar en la gestión del stock es la localización de las tiendas. Básicamente, el posicionamiento de una tienda se lleva a cabo de manera estratégica teniendo en cuenta el entorno en el que se pretende que esté localizada. Como consecuencia, el inventario de dicha tienda deberá adaptarse a los segmentos de clientes que conformen la mayoría de la clientela. Por ejemplo, si una tienda se encuentra cerca de una universidad, las tallas que deberán tener mayores niveles dentro de la curva de tallas serán aquellas que sean más pequeños debido a que el segmento predominante estará formado por universitarios.

En cuanto al stock de seguridad de las tiendas en períodos previos a temporada de rebajas, será necesario fijar unos niveles que respetan dicha estacionalidad con el fin de abastecer lo mejor posible la demanda en esas épocas de ventas masivas. Por el contrario, una vez que el período

de promociones finalice, el stock de seguridad será eliminado pasando a ser inventario que urge venderse debido a que son artículos sobrantes y que corren el riesgo de acumularse generando sobrecostes.

Para el abastecimiento de los pedidos online, se tienen en cuenta muchos más factores debido a la sensibilidad que existe de alterar el inventario de las tiendas con el surtido de los pedidos. Es necesario aclarar que los factores que se van a comentar se tienen en cuenta en el caso en el que los pedidos de la tienda online se abastecen con el inventario de las tiendas debido a que en muchos casos existen almacenes específicos para la gestión del stock de la tienda online.

En primer lugar, los criterios de cercanía y disponibilidad son la base de la gestión debido a que son aquellos que generan un mayor ahorro sobre los costes de venta de un producto. Los pedidos online deberán ser surtidos por aquella tienda que se encuentre más cerca del punto de destino, mientras que se cumplan el resto de criterios a tener en cuenta.

Otro criterio relevante en el surtido de los pedidos es que no se rompa el stock de seguridad definido para cada una de las tiendas. Dependiendo del nivel de ventas de cada una, tendrán asignadas un stock de seguridad con el fin de cubrir la demanda prevista. Por lo tanto, lo que se pretenderá es no romper dicho stock con el objetivo de no alterar la gestión del inventario de la red. Por ejemplo, aquellas tiendas consideradas como *flagship* o tiendas escaparate tendrán unos niveles de inventario de seguridad que permitan llevar a cabo su principal función, que es la de vender la imagen de la firma mostrando a los clientes todo el catálogo de prendas de la temporada. Como el objetivo es crear una experiencia al cliente con estas tiendas, se tiene que tener fijado unos niveles de stock de seguridad que no pongan en riesgo la estrategia que la compañía tiene con esa tienda en particular [HODS17].

Directamente relacionado con el stock de seguridad, existe un factor que también debe ser considerado y que, por el contrario, muchos negocios no lo tienen en cuenta. Este criterio consiste en mantener la curva de tallas intacta, no corriendo el riesgo de alterarla con el abastecimiento de los pedidos online. La curva de tallas va acorde con las ventas, por lo tanto, es fundamental mantenerla para evitar, de esta manera, la posibilidad de aumentar los niveles de venta perdida en tienda.

Como se puede apreciar, los factores comentados anteriormente no solo pretenden optimizar la gestión de la red de distribución a nivel económico, sino que con su aplicación la intención también es la de incrementar el rendimiento y la eficiencia de la red. La eficiencia está directamente relacionada con la satisfacción del cliente, aspecto clave en la fidelización que se busca en las compañías del sector.

6. Inteligencia Artificial

6.1. Definición

La Inteligencia Artificial (IA) o también llamada inteligencia computacional, se define como la rama de la ciencia de la computación que permite hacer de las máquinas un agente racional flexible que es capaz de percibir su entorno y llevar a cabo acciones que incrementen la optimización y efectividad de alguna tarea objetivo. En otras palabras, tal y como definió John McCarthy en el año 1956: “Se puede entender la inteligencia artificial como aquellos algoritmos que se materializan en programas informáticos que, a su vez, corren en un hardware determinado, y que persiguen imitar el modo de funcionamiento del cerebro humano”.

Para Nils John Nilsson (investigador fundador de la disciplina de la inteligencia artificial de la Universidad de Stanford) son cuatro los pilares básicos en los que se fundamenta la inteligencia artificial:

- Búsqueda del estado óptimo entre los distintos estados posibles definidos por las acciones posibles.
- Algoritmos genéticos.
- Redes neuronales artificiales.
- Razonamiento mediante una lógica formal.

Todos estos pilares están directamente relacionados con la forma de actuar y pensar del ser humano. Por este motivo, coloquialmente se dice que la inteligencia artificial se aplica cuando una máquina imita las funciones cognitivas que un ser humano es capaz de ejecutar, como por ejemplo aprender o resolver problemas.

Según Stuart Russel (profesor de ciencias de la computación de la Universidad de California, Berkeley) y Peter Norvig (consejero de la American Association for Artificial Intelligence), existen varios tipos de inteligencia artificial:

- Sistemas que piensan como humanos. Sistemas que tratan de emular el pensamiento humano mediante la automatización de actividades como la toma de decisiones o el aprendizaje.
- Sistemas que actúan como humanos. Imitan el comportamiento del ser humano, como por ejemplo la robótica.
- Sistemas que piensan racionalmente. Basados en el estudio de los cálculos que hacen posible percibir, razonar y actuar. Por ejemplo, los sistemas expertos que son sistemas computacionales que emulan la capacidad de toma de decisiones de un humano experto.
- Sistemas que actúan racionalmente. Sistemas que tratan de emular el comportamiento del ser humano de forma racional, como por ejemplo los agentes inteligentes, que son entidades capaces de percibir su entorno, procesar las percepciones y responder o actuar de manera

racional.

Actualmente, es un hecho que la inteligencia artificial es una de las mejores inversiones de cara al futuro. Según un estudio llevado a cabo por la consultora PwC, estiman que para el año 2030, la inteligencia artificial hará que el PIB mundial aumente a 15,7 billones de dólares, lo que supondría un crecimiento de un 14% sobre las proyecciones iniciales. Este crecimiento se ve reflejado en el gráfico mostrado en la Figura 16, donde se aprecia cómo han evolucionado las inversiones en proyectos de inteligencia artificial en los últimos cuatro años [TORR17].

Su potencial hace que empresas como Netflix, Facebook o Amazon hayan decidido apostar e invertir fuertemente por el desarrollo de la misma en sus empresas ya que consideran que les puede ser de gran ayuda para ampliar su mercado, llegar a otros sectores y mercados o posicionar sus productos entre muchas de las ventajas que esta tecnología puede llegar a aportar.

A día de hoy, China y Estados Unidos son los dos mercados a nivel mundial más importantes al ser ambos propulsores por excelencia de Inteligencia Artificial. Existen estimaciones de que para el año 2030 ambos países tendrán el 68% de la producción de esta tecnología [TORR17].

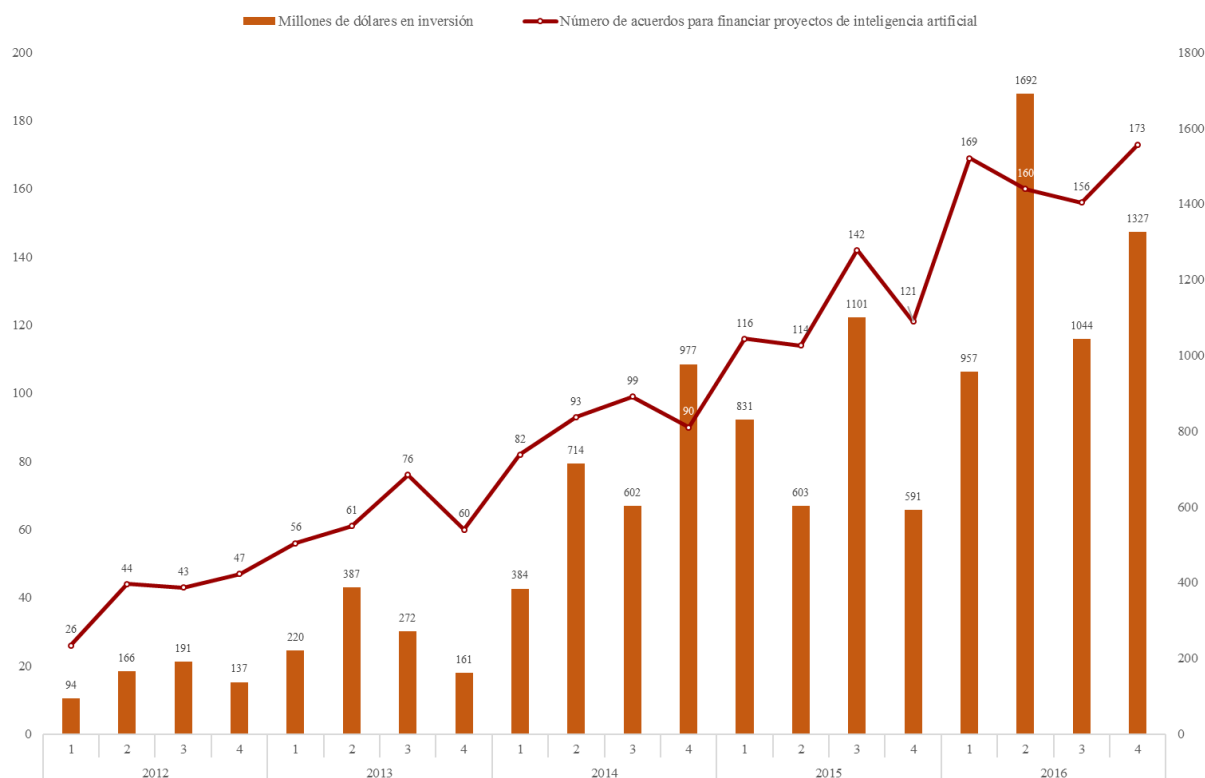


Figura 16. Inversiones en proyectos de Inteligencia Artificial. Fuente: McKinsey & Co., Accenture y CB Insights.

6.2. Machine Learning

También llamado aprendizaje automático, es una rama de la inteligencia artificial que tiene por objetivo generalizar comportamientos a partir de información en forma de ejemplos. En otras

palabras, es un proceso de inducción al conocimiento mediante la creación de programas. Puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos.

Existen, fundamentalmente, tres tipos de modelos que surgen como resultado de la aplicación del aprendizaje automático:

- Modelos geométricos. Se construyen en el espacio de instancia y pueden tener una, dos o varias dimensiones.
- Modelos probabilísticos. Aquellos cuyo objetivo es determinar la distribución de probabilidades de la función que enlaza a los valores de las características con valores determinados.
- Modelos lógicos. Transforman y muestran las probabilidades en reglas organizadas. Esto se lleva a cabo mediante árboles de decisión.

Machine Learning se fundamenta en una serie de algoritmos que, principalmente, se clasifican en dos tipos. Por un lado, aprendizaje supervisado, que son aquellos que establecen una relación entre las entradas y las salidas a través de una función. Cuenta con información que especifica qué conjuntos de datos son satisfactorios para el objetivo de aprendizaje. Por otro lado, el aprendizaje no supervisado, en el que el sistema tiene que ser capaz de reconocer patrones para poder clasificar los datos de las entradas del sistema. La información de salida, fruto de la ejecución del modelo, debe de ser interpretada por una persona posteriormente para que resulte útil [RAO_16].

En general, los modelos más característicos del aprendizaje automático son clasificación, regresión, redes neuronales artificiales, árboles de decisión, etc.

Esta tecnología actualmente es crucial en el día a día de las empresas debido a que la cantidad de datos que se generan se está incrementando de forma exponencial. Por lo tanto, tener la capacidad de extraer información significativa de ellos supone una ventaja competitiva que no se debe de pasar por alto. Con datos de calidad el resultado es la obtención de unas predicciones de alto valor que permiten tomar decisiones más acertadas y desarrollar mejores acciones de negocio.

6.3. Inteligencia Artificial en el sector retail

Es un hecho que la Inteligencia Artificial (IA) formará parte del futuro inmediato de las empresas del sector retail y de los consumidores. Ciertamente, ya hay empresas pioneras que están incorporando el uso de dicha tecnología dentro de sus estrategias digitales. De hecho, es algo tan común en plena Transformación Digital, que los ciudadanos de a pie ya son conocedores de los beneficios de estas técnicas.

La Inteligencia Artificial permite a las compañías del sector dar respuesta a muchos de sus problemas más urgentes, como son:

- **Ubicación del comercio.** Las decisiones de este tipo se llevaban a cabo hasta ahora mediante análisis predictivo, sin embargo, este aspecto se puede ya optimizar con la ayuda de esta tecnología. Con la Inteligencia Artificial se puede, además de determinar el lugar y el momento idóneo de apertura de una tienda, obtener una comprensión del porqué de esa decisión, identificando los impulsores y variables críticas que contribuyen en dicha recomendación.
- **Dotación de personal.** Con ayuda de esta tecnología las compañías pueden determinar el número de empleados necesarios que deben de tener en sus tiendas tanto en período regular como en épocas de rebajas. En el caso de los períodos de rebaja, los minoristas pueden conocer con exactitud el número de empleados extra necesarios para cada día y en qué áreas habrá mayor necesidad. El resultado de esta toma de decisión tendrá un fuerte impacto en el balance de la empresa reduciendo los costes y proveyendo una mejor experiencia en la tienda para los clientes.
- **Gestión de inventario.** Este aspecto es elemental para las empresas del sector debido a que influye significativamente sobre los flujos de ingresos. Para evitar errores o desequilibrios en el inventario será necesario el uso de modelos de predicción que permitan determinar qué artículos y dónde serán requeridos. Para poder obtener esos datos, se deben tener en cuenta una serie de factores como por ejemplo la estacionalidad, ubicación, etc. Para resolver este asunto, en el cual hay tantas variables involucradas, es imprescindible el uso de Inteligencia Artificial.
- **Digitalizar las tiendas.** Los comercios tienen la necesidad de digitalizar al cliente en la tienda física con el objetivo de establecer la integración progresiva de ambos canales y, como consecuencia, generar experiencias de compras más enriquecedoras. Solamente digitalizando este aspecto se conseguirá incorporar de forma natural la Inteligencia Artificial orientada al cliente.

Además de los aspectos mencionados anteriormente, la Inteligencia Artificial puede ser de gran utilidad en campos como el marketing, para mejorar la efectividad de las campañas, afinar los pronósticos de venta, con objeto de mejorar la asignación de recursos, y para la contratación de empleados, evitando de esta manera la rotación y apostando por los más cualificados para el puesto y la entidad. Todos estos beneficios se muestran en la Figura 17, donde se definen las principales contribuciones que la Inteligencia Artificial aporta a las empresas del sector.

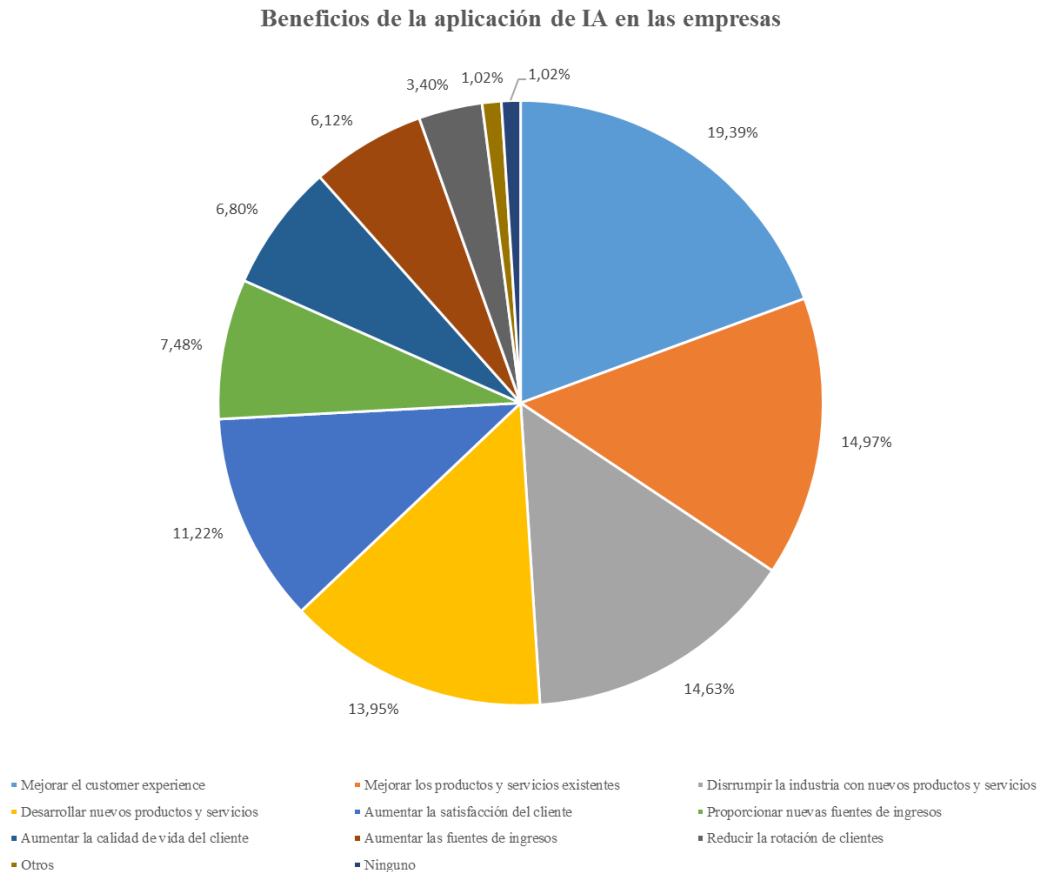


Figura 17. Principales beneficios de la IA en las empresas del sector. Fuente: Digiday UK.

Los líderes del sector retail están actualmente incorporando estas técnicas a sus modelos de negocio. Amazon está desarrollando e implantando algoritmos de *machine learning* con el fin de analizar y crear nuevas tendencias de moda, Walmart ha incorporado a su aplicación móvil la funcionalidad “escanea y compra”, que permite a los consumidores evitar grandes colas. Por otro lado, Levi’s ha implementado en su página web *chatbots* con el fin de ayudar a los clientes a encontrar sus pantalones perfectos. Con esta estrategia lo que pretenden es mejorar la experiencia del cliente a la vez que implementan un servicio de personalización, dos factores determinantes que se encuentran dentro de los principales requerimientos de los consumidores actuales [BROW16].

A parte de las implementaciones mencionadas, muchos retailers están incorporando mejoras en sus canales de comunicación con sus clientes que permiten la personalización de los mismos en tiempo real, generando mejores recomendaciones de productos o estableciendo alternativas más efectivas e inmediatas de respuesta por parte de sus centros de atención al cliente, siempre desde el punto de vista de la digitalización.

Un aspecto complejo en estos procesos de transformación digital es la integración. La dificultad no reside en la integración de tecnologías una a una, sino en la integración de ecosistemas completos a la misma vez que estos crecen como tal. Las empresas del sector retail presentan

arquitecturas sobre las cuales las empresas están sumergidas en el día a día. Este aspecto hace que la implementación tenga que hacerse en paralelo por equipos especializados para poder alcanzar la agilidad de proceso pretendida.

Con vistas al futuro de esta tecnología, hay un amplio margen de evolución y mejora en este sector. Uno de los puntos con mayor potencial relacionado directamente con el consumidor es la identificación del cliente omnicanal donde se pretende vincular al cliente que normalmente compra en tienda física con el cliente digital.

6.4. Data Analytics en el sector Retail

Las tecnologías de análisis de datos han penetrado en el sector retail consolidándose como una herramienta fundamental de trabajo que permite mejorar el rendimiento de la compañía a través de un gran volumen y calidad de datos preparados para ser analizados.

Una de las principales ventajas que esta tecnología aporta al sector es la posibilidad de conocer los movimientos de la competencia con el fin de poder anticiparse a sus estrategias. Como se comentaba en el apartado de introducción del proyecto, este hecho hace que las posibilidades de destacar y de liderar el sector sean cada vez más complejas. Por otro lado, la aplicación de las técnicas de análisis de datos permite conocer al cliente mejor, sabiendo su comportamiento de cara a la compra y sus tendencias. Con ello, lo que se pretende es ofrecerles un servicio cada vez más personalizado. Por último, otra ventaja que ofrece es la capacidad de adelantarse a cambios del mercado.

Con la incorporación de esta tecnología a las empresas del sector se han apreciado una serie de cambios en la forma de gestionar los negocios que han sido fruto de los análisis que el Data Analytics permite llevar a cabo.

- Modificación del layout de los productos en las tiendas con el objetivo de que el cliente recorra la misma al completo. De esta forma, lo que se pretende es establecer estrategias para potenciar la venta de una serie de productos que, de otra forma, no resultan tan atractivos.
- Previsiones de venta mediante la ejecución de análisis estadísticos.
- Gestión del inventario y mejora del flujo del suministro de productos.
- Análisis de la fidelidad del cliente a través de los datos que proporciona el CRM. Este aspecto es muy importante ya que la búsqueda de nuevos clientes resulta mucho más costosa que mantener a los clientes existentes. Por lo tanto, hay un gran esfuerzo por parte de las empresas por mejorar la experiencia de sus clientes.
- Optimizar la estrategia de pricing.

Como se puede ver, las aplicaciones mencionadas anteriormente se llevan a cabo en cada una de las etapas de la cadena de suministro de cualquier empresa del sector retail.

Unos de los aspectos claves en el análisis de datos son las promociones, descuentos u ofertas. Las empresas a través de análisis pueden ver si las promociones que lanzan son efectivas. Esto se lleva a cabo mediante la recolección de datos históricos y posterior uso en algoritmos de predicción que permitan conocer qué impacto podrá tener la estrategia que están ejecutando a largo plazo. Como solución, pueden identificar las ofertas que aportan un valor añadido al cliente o, al contrario, eliminar aquellas que no hayan tenido éxito.

Empresas gigantes del sector como por ejemplo Walmart invierten grandes cantidades de dinero en la adquisición de estructuras de datos privados para poder soportar millones de transacciones que experimentan diariamente y, sobre ellas, poder ejecutar funciones de análisis a través de algoritmos.

Un claro ejemplo de aplicación de Data Analytics en uno de los mayores retailers de la actualidad es el caso de Costco. Esta compañía a través de la recopilación exhaustiva de datos de compra de todos sus clientes es capaz de identificar los clientes potenciales en función de cada uno de los productos que ofertan y de mandar avisos que advierten de situaciones concretas. Por ejemplo, en California se produjo una contaminación de listeria en frutas como nectarinas y melocotones. Como medida preventiva, haciendo uso de los análisis de datos realizados, mando mensajes de aviso vía telefónica y por carta a los clientes potenciales de estos productos advirtiéndoles de la situación.

Otro caso de éxito está relacionado con la plataforma meteorológica The Weather Channel. La mayoría de las personas piensan que ésta sólo está relacionada con la predicción del clima pero en realidad va mucho más allá que eso. A través de sus plataformas de datos, Location FX y Weather FX, el canal The Weather monitoriza el impacto del clima en las emociones de los consumidores. Estos análisis meteorológicos predictivos analizan las tendencias según la ubicación y guían a los anunciantes sobre cómo y cuándo enviar su mensaje o anunciar su campaña o promoción con el fin de causar mayor impacto en los clientes.

Un ejemplo de esto fue la asociación entre Pantene, Walgreens y Weather Channel. Utilizando los datos recopilados por The Weather Channel, Pantene y Walgreens pudieron anticipar cuándo la humedad en el aire alcanzaría su punto más alto, lo que llevó a las mujeres a buscar un producto en su farmacia local para evitar el encrespamiento y el cabello despeinado .

Esto fue calificado como un "haircast" y condujo a un aumento del 10% en las ventas de Pantene en Walgreens durante los meses de julio y agosto, junto con un aumento de ventas del 4% en toda la categoría de cuidado del cabello en Walgreens. También estimuló la creación de discusiones en las redes sociales bajo la etiqueta #haircast [MOBI16].

Estos casos de uso del Big Data en el sector retail son un reflejo de la evolución de esta tecnología y de su situación en la actualidad. Es un hecho que el análisis de datos está cada vez más incorporado a la operativa de las empresas y que, conforme va evolucionando, hay un mayor número de especialistas en la materia. Sin embargo, el poseer los datos y analizarlos no es suficiente, sino que lo que realmente define una estrategia de diferenciación es el modo en el que se enfoca el análisis de los datos y con qué fin.

7. Desarrollo del modelo de optimización

7.1. Introducción

El primer modelo que se pretende desarrollar en este proyecto es un modelo de base analítica cuyo objetivo es el de optimizar la red de distribución de una compañía tipo del sector retail. Para llevar a cabo el desarrollo del modelo se ha empleado el lenguaje de programación Python, con el cuál se han generado una serie de lógicas tanto de generación como de simulación de un sistema de gestión de pedidos.

Para poder hacer uso del modelo que se va a implementar, se ha de definir previamente la red de distribución y una base de datos de la compañía tipo sobre la que se pretende trabajar. Para ello, se ha usado un modelo de base de datos relacional donde se definen los artículos que conforman el stock, los clientes, las tiendas y almacenes, etc.

Se ha optado por implementar una base de datos relacional debido a que es el modelo más utilizado actualmente en la elaboración de bases de datos ya planificadas. Esta tipología permite la integración de todas las tablas que conforman la base de datos mediante claves primarias que relacionan una tabla padre con las hijas. El motivo de haber optado por este modelo es la facilidad que proporciona tanto a la hora de trabajar con los datos como para la posterior representación de los resultados.

Por otro lado, en relación con el desarrollo del modelo de gestión de pedidos, lo que se pretende es plantear una serie de escenarios en los que se tengan en cuenta distintos criterios de gestión, dentro de los mencionados en el *Apartado 5.2.5: Factores influyentes en la gestión de una red de distribución*.

Con la definición de las características de cada uno de los escenarios, el siguiente paso será la simulación de la gestión de los pedidos, obteniendo por un lado el resultado de la distribución de las unidades que conforman los pedidos a lo largo de la red establecida y, por otro lado, una valoración económica de los costes operativos, de mercancías e ingresos obtenidos en la simulación llevada a cabo.

Una vez obtenido los resultados de todos los escenarios tanto a nivel operativo como a nivel económico, se llevará a cabo una ponderación de todos ellos con el fin de valorar cuál se ajusta mejor a las necesidades actuales de las empresas del sector.

7.2. Diseño de la red de distribución

La configuración de la red de distribución que se va a emplear en el desarrollo del modelo varía en función de los escenarios que se van a plantear, a causa de la operativa de servicio de pedidos online que es, fundamentalmente, el punto sobre el que pivotan cada uno de los escenarios que se presentarán en apartados posteriores.

A nivel general, la red de distribución está compuesta por un centro de almacenamiento, a través del cual se distribuyen los artículos para la reposición de las tiendas que conforman la red a nivel territorial. Este almacén está situado en la localidad de Toledo, punto estratégico para establecer la logística de manera más eficiente debido a que está aproximadamente en el punto medio del territorio sobre el que se ha definido la red.

Por otro lado, hay un total de 14 tiendas repartidas por todo el país. Principalmente, se encuentran situadas en las ciudades más significativas y algunas de ellas en otras ciudades que han sido designadas para que la red de distribución tenga mayor extensión. Los datos geográficos del almacén y de cada una de las tiendas se presentan a continuación en la Tabla 1 y Tabla 2 respectivamente. A parte, en la Figura 18 se muestra un mapa de la red definida.

ID Almacén	Almacén	Ciudad	Provincia	Localización	Código Postal	Latitud	Longitud
1	Almacén	Toledo	Toledo	Calle de Malamata, 5	28830	40,0491	-3,9151

Tabla 1. Datos geográficos del almacén.

ID Tienda	Tienda	Ciudad	Provincia	Localización	Código Postal	Latitud	Longitud
1	Tienda 1	Santiago de Compostela	La Coruña	Rúa de Malta, 5	15704	42,89959	-8,509760
2	Tienda 2	Santander	Santander	Calle Joaquín Rodrigo, 15	39011	43,45641	-3,860819
3	Tienda 3	Pamplona	Pamplona	Calle Universidad, 23	31191	42,79989	-1,649484
4	Tienda 4	Barcelona	Barcelona	Avenida Diagonal, 450	8036	41,39643	2,158147
5	Tienda 5	Barcelona	Barcelona	Paseo de Gracia, 85	8007	41,39431	2,161555
6	Tienda 6	Barcelona	Barcelona	Avenida del Portal del Ángel, 21	8002	41,38567	2,172347
7	Tienda 7	Salamanca	Salamanca	Calle Capuchinos, 1	37900	40,94945	-5,650446
8	Tienda 8	Madrid	Madrid	Plaza del Carmen, 6	28013	40,41859	-3,703444
9	Tienda 9	Madrid	Madrid	Calle Azafrán, 15	28023	40,45009	-3,766117
10	Tienda 10	Madrid	Madrid	Gran Vía, 102	28013	40,42322	-3,710584
11	Tienda 11	Madrid	Madrid	Calle Serrano, 85	28006	40,43668	-3,686579
12	Tienda 12	Sevilla	Sevilla	Avenida Kansas City, 43	41007	37,39565	-5,968340
13	Tienda 13	Cartagena	Murcia	Calle Londres, 85	30392	37,62059	-0,996935
14	Tienda 14	Valencia	Valencia	Avenida Cortes Valencianas, 31	46015	38,95733	-0,588362

Tabla 2. Datos geográficos de las tiendas.



Figura 18. Mapa de la red de distribución.

Como se aprecia en la Tabla 2, en las ciudades con mayor población y, a su vez, con mayor turismo se han definido varias tiendas distribuidas por las zonas más conocidas de cada una de las respectivas ciudades. Con esto se pretende seguir los criterios que las empresas del sector emplean para establecer su estrategia de apertura de nuevas tiendas físicas. En general, pretenden cubrir los niveles máximos de población, situando las tiendas en ciudades geográficamente importantes en el mapa y, también, teniendo en cuenta factores como el turismo y los niveles de renta per cápita de las zonas de estudio en función de la segmentación que pretendan llevar a cabo.

Como característica adicional, se han definido que las tiendas número 5 y 11 serán catalogadas como *flagship*, es decir, que su principal función será la de hacer de tiendas escaparates de cara a los clientes. Esto afectará en variables como los niveles de inventario inicial o el stock de seguridad, que serán definidos en apartados posteriores de la memoria. Sin embargo, el resto de tiendas serán tiendas tradicionales cuyo objetivo primario será el de alcanzar los mayores niveles de venta posible.

Para cada una de las tiendas se han definido una serie de parámetros propios del sector que se mostrarán en el apartado siguiente de la memoria. Con esto lo que se pretende es que haya

diversidad dentro de la red de distribución y que, de esta forma, se pueda alcanzar un nivel de realismo del modelo apropiado.

Como se ha mencionado antes, el objetivo fundamental del almacén es el llevar a cabo la reposición de las tiendas, de ahí su posicionamiento en el mapa. Sin embargo, en el primer escenario se simulará el abastecimiento de los pedidos online mediante el stock de dicho almacén, siguiendo la operativa de muchas empresas del sector que establecen un almacén concreto para cubrir la demanda de la tienda online. Con esta estrategia impuesta, el posicionamiento del almacén también sería correcto ya que es equidistante a la mayoría de las tiendas situadas en los extremos del mapa.

7.3. Definición de la base de datos

La base de datos empleada para la elaboración y simulación del modelo de gestión de pedidos que se va a definir se corresponde con una base de datos del sector de moda, dentro del sector retail. El modelo usado es de tipo relacional. Este tipo de base de datos se caracteriza por la integración que existe entre cada uno de los elementos que la conforman. Las formas relacionales entre las tablas se llevan a cabo mediante claves. Por un lado, están las claves primarias, que son las principales y las que se usan en las tablas padre, las que deben de asegurar la integridad entre las tablas. Además, son las que definen unívocamente a todos los demás atributos de la tabla que guardan una relación con otras tablas distintas. Por otro lado, están las claves ajenas o foráneas que se colocan en las tablas hijas y que contienen el mismo valor que la clave primaria del registro padre.

Para comenzar con la definición de la base de datos de este proyecto, lo primero es mencionar que esta está compuesta por un total de 10 tablas. Por un lado, están las tablas que definen los artículos que conformarán el inventario del negocio definido y, por otro lado, estarán las tablas que hacen referencia a los elementos de la red de distribución, es decir, tiendas, almacenes y clientes. A parte, estarán las tablas relacionadas con los pedidos, devoluciones y stock inicial que serán generadas en los próximos apartados ya que forman parte de los pasos previos del desarrollo del modelo.

Básicamente, las tablas que no definen los artículos y que conforman la base de datos son las mostradas a continuación:

- Tiendas
- Almacenes
- Clientes
- Pedidos
- Devoluciones

- Stock Inicial

Por otro lado, las tablas que definen los artículos son:

- Familias
- Subfamilias
- Marcas
- Artículos

La tabla de tiendas y la de centros han sido definidas previamente en el apartado anterior. Se han comentado los parámetros relacionados con la posición geográfica de cada uno de los centros que conforman la red de distribución. Sin embargo, en el caso de la tabla de tiendas hay una serie de parámetros que también se han definido en dicha tabla y que permiten llevar a cabo la caracterización individual de cada una de las tiendas. Las columnas restantes no definidas en el apartado anterior se muestran a continuación en la Tabla 3.

ID Tienda	Tienda	Ticket Medio	UPT	Ventas totales mensuales	Porcentaje de devoluciones	Porcentaje Stock	Stock de seguridad
1	Tienda 1	77,6	1,6	27603,4	0,1	1,1	0
2	Tienda 2	116,5	1,8	29999,3	0,1	1,1	1
3	Tienda 3	87,9	2,1	30230,6	0,2	1,2	0
4	Tienda 4	112,9	1,7	29231,9	0,1	1,1	1
5	Tienda 5	150,4	1,5	44578,0	0,2	1,4	4
6	Tienda 6	124,1	2,3	32156,5	0,2	1,0	1
7	Tienda 7	110,1	1,6	27518,0	0,1	1,1	1
8	Tienda 8	93,9	2,0	33067,4	0,1	1,1	1
9	Tienda 9	91,5	1,8	32637,5	0,1	1,2	0
10	Tienda 10	139,5	2,4	37125,5	0,2	1,0	1
11	Tienda 11	145,4	1,5	46780,0	0,2	1,4	4
12	Tienda 12	118,0	1,9	31798,8	0,1	1,0	1
13	Tienda 13	80,9	1,7	27168,1	0,1	1,2	1
14	Tienda 14	131,5	1,9	30698,5	0,1	1,1	0
15	Tienda 15	75,6	1,7	125460,0	0,2	0,6	0

Tabla 3. Tabla de datos de tiendas. Parámetros retail.

- **Ticket medio.** Es el valor medio de la cantidad que gasta cada cliente en una compra.
- **UPT (Unidades por Ticket).** Es muy similar al ticket medio. Es el número medio de artículos que cada cliente adquiere en una compra.
- **Ventas totales mensuales.** Es el valor medio mensual de ingresos para cada una de las tiendas.
- **Porcentaje de devoluciones.** Es el valor porcentual de devoluciones que una tienda tiene sobre el número de pedidos que ha recibido en el tiempo de simulación definido.

- **Porcentaje de stock.** Este parámetro es uno de los parámetros de entrada del generador de inventario inicial. Con él se definirá el exceso o defecto de stock de cada una de las tiendas.
- **Stock de seguridad.** Es el número de unidades por cada artículo del inventario que conforman el stock de seguridad para cada una de las tiendas. Como se puede apreciar, las tiendas definidas como *flagship* son aquellas que mayor stock de seguridad tienen debido a que tienen que asegurar la existencia de todo el catálogo de artículos en la tienda para poder acometer su principal función, que es la de tienda de exposición.

Es necesario comentar que todos estos parámetros han sido previamente a la construcción del modelo ya que serán de gran ayuda en la generación de los pedidos, las devoluciones y el stock inicial.

A parte, es fundamental destacar que se ha definido el valor de las ventas mensuales para la tienda online cerca de tres veces el de una tienda física. El motivo de esta decisión es la necesidad de aportar dinamismo al modelo para poder obtener resultados claros, evitando posibles confusiones.

Con respecto a los clientes, se han definido un total de 100 clientes, separando aquellos que llevan a cabo compras en tienda física y aquellos que solicitan pedidos a través de la tienda online. Se han definido una serie de características que también serán fundamentales para la generación de los pedidos. Por lo tanto, para presentar de forma breve la tabla, se muestra en la Tabla 4 las primeras líneas de la misma.

Id Cliente	Cliente	Género	Latitud	Longitud	Ticket medio	UPT
1	Álvaro Garcia Méndez	H	42,89960	-8,50976	64,00	1,8
2	Vicente González Cruz	H	43,45641	-3,86082	124,20	2,7
3	Aurora Rodríguez Calvo	M	42,79990	-1,64948	113,00	1,1
4	Ángeles Fernández Gallego	M	40,28709	-3,60960	163,30	1
5	Patricia López Vidal	M	41,39431	2,16156	115,00	3

Tabla 4. Extracto de la tabla de clientes.

Como se ha mencionado anteriormente, hay una serie de parámetros, que son el género, el ticket medio y UPT, que son claves para la generación de pedidos debido a que para el caso del género, este permite identificar la familia de productos a los cuales puede el cliente acceder y realizar una compra o un pedido.

En cuanto a la situación geográfica de cada cliente, la manera en la que se ha identificado a aquellos clientes que compran en tienda o a través de la plataforma e-commerce es a través de este parámetro. Aquellos clientes que compran en tienda, tendrán asignada la latitud y longitud de dicha tienda. Sin embargo, aquellos que compran en la tienda online tendrán las coordenadas geográficas de su domicilio correspondiente.

Una vez comentadas las características generales de las tablas que conforman la red logística del modelo, se va a explicar el procedimiento mediante el cual se han elaborado las tablas

relacionadas con los artículos de la compañía retail. Para la definición de los artículos se ha tomado como referencia una base de datos de una compañía del sector con el fin de asemejar la base de datos lo máximo posible a una base de datos real.

El catálogo de artículos se ha definido siguiendo la estructura tradicional del sector. En primer lugar, se tienen las familias de artículos, que son básicamente tres, mostradas en la Tabla 5, y, por otro lado, las subfamilias que son 16. Las subfamilias son las correspondientes a la temporada que se procederá a simular, que es la temporada de otoño – invierno. Dichas subfamilias se muestran en la Tabla 6.

Id Familia	Familia
01	Moda Mujer
02	Moda Hombre
03	Moda Joven

Tabla 5. Familias de artículos de la BBDD.

Dentro de las familias y subfamilias, se han definido una serie de marcas estandarizadas que tienen artículos bien en varias de las familias y subfamilias, o bien son específicas para una subfamilia en concreto. Hay un total de 91 marcas, que toman el nombre estándar de M seguido del número de la marca correspondiente.

Una vez definidas las familias, subfamilias y marcas, se puede proceder a elaborar la tabla de los artículos, que tiene un total de 779 artículos en total con un reparto equitativo de artículos entre cada una de las subfamilias. La Tabla 7, muestra un extracto de la tabla de artículos definida con un ejemplo de un artículo de cada una de las subfamilias y un ejemplo del conjunto de tallas completo de un artículo padre.

Id Subfamilia	Subfamilia
01	Abrigos
02	Camisas, blusas, tops
03	Camisetas
04	Cazadoras
05	Chalecos
06	Chaquetas y blazers
07	Faldas
08	Pantalones
09	Polos
10	Sudaderas
11	Vaqueros
12	Vestidos
13	Americanas y blazers
14	Camisas
15	Ropa interior y calcetines
16	Trajes

Tabla 6. Subfamilias de artículos.

SKU	ID Familia	ID Subfamilia	ID Marca	ID Artículo	Artículo	Dim. paquete	Talla	Precio
000011231	1	01	23	0001	Parka Marca M23 Talla XS	G	XS	34,96 €
000011232	1	01	23	0002	Parka Marca M23 Talla S	G	S	34,96 €
000011233	1	01	23	0003	Parka Marca M23 Talla M	G	M	34,96 €
000011234	1	01	23	0004	Parka Marca M23 Talla L	G	L	34,96 €
000128149	1	02	81	0049	Top Marca M81 Talla XS	P	XS	29,95 €
000133685	1	03	36	0085	Camiseta Marca M36 Talla XS	P	XS	24,95 €
001434110	1	04	34	0110	Cazadora Marca M34 Talla S	M	S	169,95 €
001536141	1	05	36	0141	Chaleco Marca M36 Talla XS	P	XS	41,95 €
001657159	1	06	57	0159	Chaqueta Marca M57 Talla M	M	M	95,50 €
001732178	1	07	32	0178	Falda Marca M32 Talla S	P	S	29,99 €
001882203	1	08	82	0203	Pantalón Marca M82 Talla M	P	M	139,00 €
002958434	2	09	58	0434	Polo Marca M58 Talla M	P	M	89,00 €
011034223	1	10	34	0223	Sudadera Marca M34 Talla M	P	M	119,95 €
031152762	3	11	52	0762	Vaquero Marca M52 Talla M	P	M	75,00 €
011262259	1	12	62	0259	Vestido Marca M62 Talla M	P	M	44,95 €
021331303	2	13	31	0303	Americana Marca M31 Talla L	M	L	225,00 €
021426321	2	14	26	0321	Camisa Marca M26 Talla S	P	S	39,95 €
031523753	3	15	23	0753	Boxer Marca M23 Talla L	P	L	7,95 €
021620490	2	16	20	0490	Traje Marca M20 Talla M	G	M	153,30 €

Tabla 7. Artículos.

Los principales parámetros que son necesarios comentar son los mostrados a continuación:

- **SKU (*Stock – keeping unit*)**. Es el número de referencia de cada uno de los artículos. La numeración se ha llevado a cabo concatenando los ID de las familias, subfamilias, marca y artículo. A nivel de distribución, es fundamental esta codificación ya que permite identificar los artículos en cualquier momento, analizar patrones de compra-venta, seguimiento de precios y fluctuaciones en el inventario.
- **ID Artículo**. Los artículos están codificados siguiendo una configuración en la hay un artículo padre, que sería el general, y los artículos hijo, que serían cada una de las tallas disponibles dentro de un artículo padre. Como se puede ver en las primeras filas de la Tabla 7, cada artículo hijo tiene una codificación propia.
- **Artículos**. Breve descripción del artículo, donde se especifica la marca y la talla de manera complementaria.
- **Dimensión paquete**. Permite identificar el tamaño del paquete necesario que habría que emplear en el caso de tener que hacer el *packaging* de dicho artículo debido a que ha sido

solicitado a través de la tienda online. Hay tres tamaños: pequeño, mediano y grande.

Con el fin de poder cerrar la base de datos relacional, es necesaria la creación de una tabla que recoja la jerarquía existente a nivel de artículos. Es decir, partiendo del nivel inferior, que sería el ID Artículo, establecer una relación de cada una de las codificaciones acabando en el SKU, que sería el código final del artículo.

Una vez definida la base de datos de los elementos de la red logística y de los artículos, se puede llevar a cabo la generación de los pedidos, devoluciones y stock inicial con mayor facilidad debido a que la codificación realizada permite trabajar con las tablas de forma más dinámica y ágil. A través de los códigos de los artículos solicitados en los pedidos se puede acceder a datos que se encuentran dentro de las tablas de los artículos o de cualquier otro elemento de la base de datos, como los clientes o la tienda en la que se ha realizado la compra.

A continuación, en la Figura 19 se muestra un esquema de las relaciones de cada uno de los elementos de la base datos con el fin de ver de forma visual la integración existente a través del sistema de codificación impuesto.

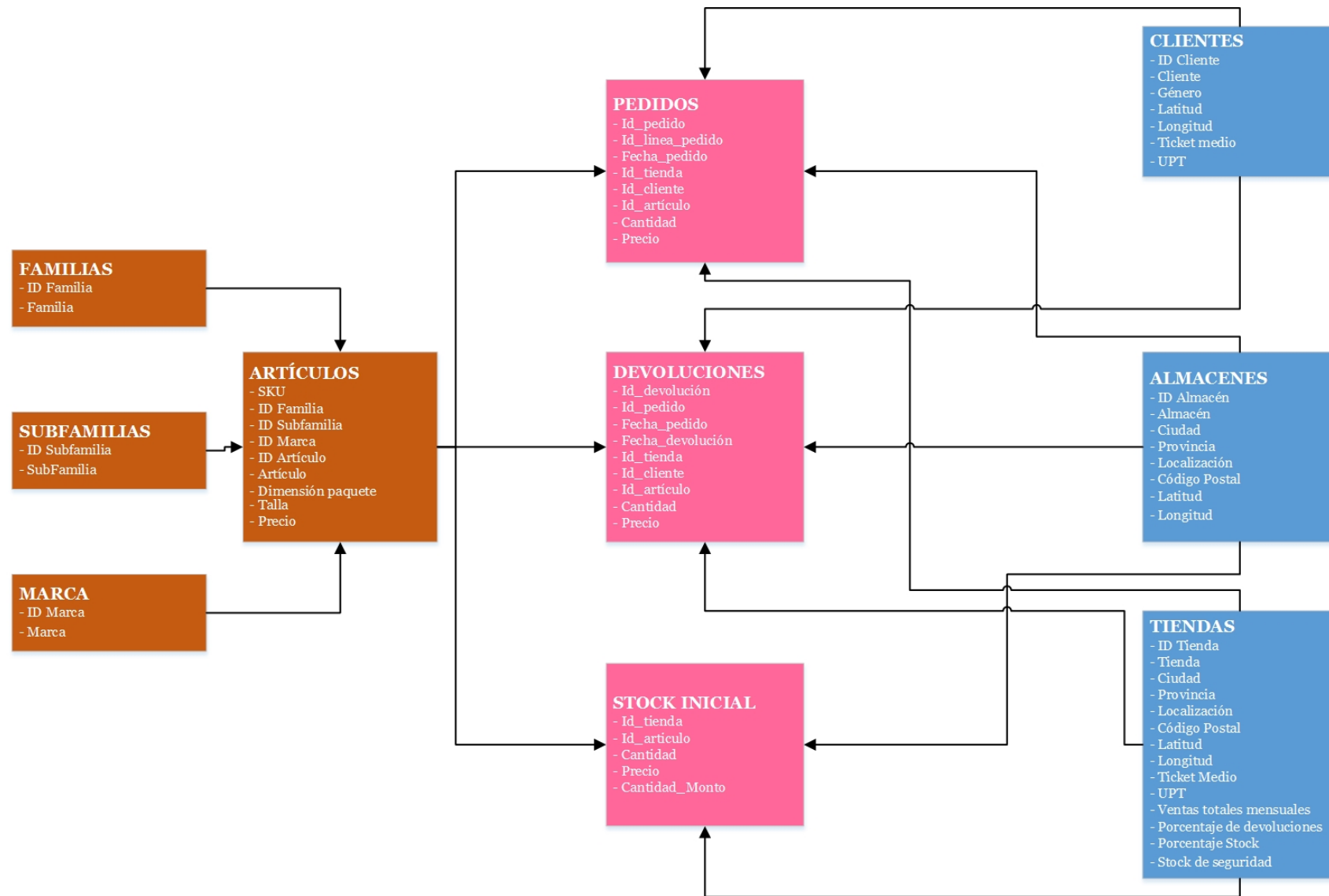


Figura 19. Representación de la integración de la BBDD.

7.4. Generador de pedidos

El generador de pedidos es una herramienta que se ha desarrollado de forma complementaria al modelo de gestión de pedidos. El motivo fundamental del desarrollo de esta herramienta es que es de vital importancia para poder generar el input del OMS. La tabla que corresponde a los pedidos que la empresa tipo del modelo recibe forma parte de la base de datos del sistema, como se mencionó en el apartado anterior.

La lógica del generador se basa en una generación aleatoria de pedidos que se reparten entre cada una de las tiendas físicas, que son tenidos en cuenta como compra física, y también en la tienda online, que serían pedidos que los clientes llevan a cabo a través de la plataforma de venta por Internet.

El input que recibe el generador es la fecha de simulación de los pedidos, es decir, el período de meses en el cuál se quiere generar pedidos para la compañía. Además, recibirá como inputs las tablas de la base de datos siguientes:

- Artículos
- Tiendas
- Clientes

El procedimiento de generación de los pedidos empieza con la selección de un cliente de forma aleatoria. Debido a que en la tabla de clientes se encuentra la localización geográfica del cliente en cuestión, se puede determinar por asignación directa cuál será la tienda a la que se realizará el pedido o compra. Hay que tener en cuenta que si la localización del cliente no coincide con la de ninguna tienda física, se le asignará como un cliente que realiza una compra online, asignando la Tienda Online definida en la tabla de la base de datos.

Una vez que se han seleccionado tanto el cliente como la tienda, habrá que seguir la lógica establecida para definir las líneas de pedido que corresponden con el pedido global que va a ordenar el cliente. Esto se determina haciendo uso de las Unidades por Ticket (UPT) del cliente correspondiente. Por ejemplo, si el cliente seleccionado tiene un UPT de 1,8 unidades, el generador decidirá de forma aleatoria si añadir una o dos líneas de pedido.

El siguiente paso, tras haber seleccionado los datos principales del pedido, es el de completar la línea de pedido con datos como la fecha del pedido, el ID, etc.

Es importante comentar que la generación de los pedidos es cronológica, es decir, el generador recibe el input del período de simulación y, una vez recibido, va generando día a día los pedidos de todas las tiendas. El parámetro que determina el cambio de día es el valor medio de ventas diaria de cada una de las tiendas, obtenido de la columna de “Ventas totales mensuales” de la tabla de Tiendas de la base de datos del sistema. El generador va realizando un recuento de los pedidos o compras que va recibiendo cada tienda y, una vez que todas las tiendas han alcanzado

su valor máximo definido, el generador comienza a generar los pedidos del siguiente día comenzando de nuevo la cuenta de los ingresos por tienda.

El resultado de la ejecución del generador de pedidos desarrollado es una tabla, que forma parte de la base de datos, que tiene el formato mostrado en la Tabla 8. Se muestra un extracto de los primeros pedidos generados debido a que el tamaño de la tabla no permite ser mostrada al completo en este documento.

ID Pedido	ID Línea Pedido	Fecha Pedido	ID Tienda	ID Cliente	ID Artículo	Cantidad	Precio
1	1	2017-10-01 15:53:00	9	45	366	1	195,00 €
1	2	2017-10-01 15:53:00	9	45	283	1	79,95 €
2	1	2017-10-01 03:39:09	4	68	388	1	249,95 €
2	2	2017-10-01 03:39:09	4	68	483	1	169,00 €
3	1	2017-10-01 21:50:36	6	84	496	1	59,99 €
3	2	2017-10-01 21:50:36	6	84	408	1	59,99 €
4	1	2017-10-01 06:24:24	10	88	482	1	169,00 €
5	1	2017-10-01 01:38:39	13	13	34	1	250,00 €
...	...	...	...	...	...	...	...
13850	1	2017-12-30 20:02:23	13	77	547	1	69,95 €
13850	2	2017-12-30 20:02:23	13	77	573	1	27,95 €

Tabla 8. Extracto de la tabla de Pedidos.

Como se puede apreciar, para el caso del modelo que se ha desarrollado en este proyecto, se ha simulado la generación de pedidos para un período de tres meses comprendido entre los meses de octubre y diciembre. Como se mencionó en el apartado anterior, la simulación se ha llevado a cabo para obtener el estudio correspondiente a la temporada de otoño – invierno.

El código correspondiente al generador de pedidos se ha desarrollado haciendo uso del lenguaje de programación Python. Dicho código se encuentra en el Anexo 1: Generador de Pedidos.

A continuación, se muestra un diagrama de flujo de la lógica utilizada en el desarrollo del generador de pedidos para facilitar, de manera visual, el entendimiento de dicha lógica.

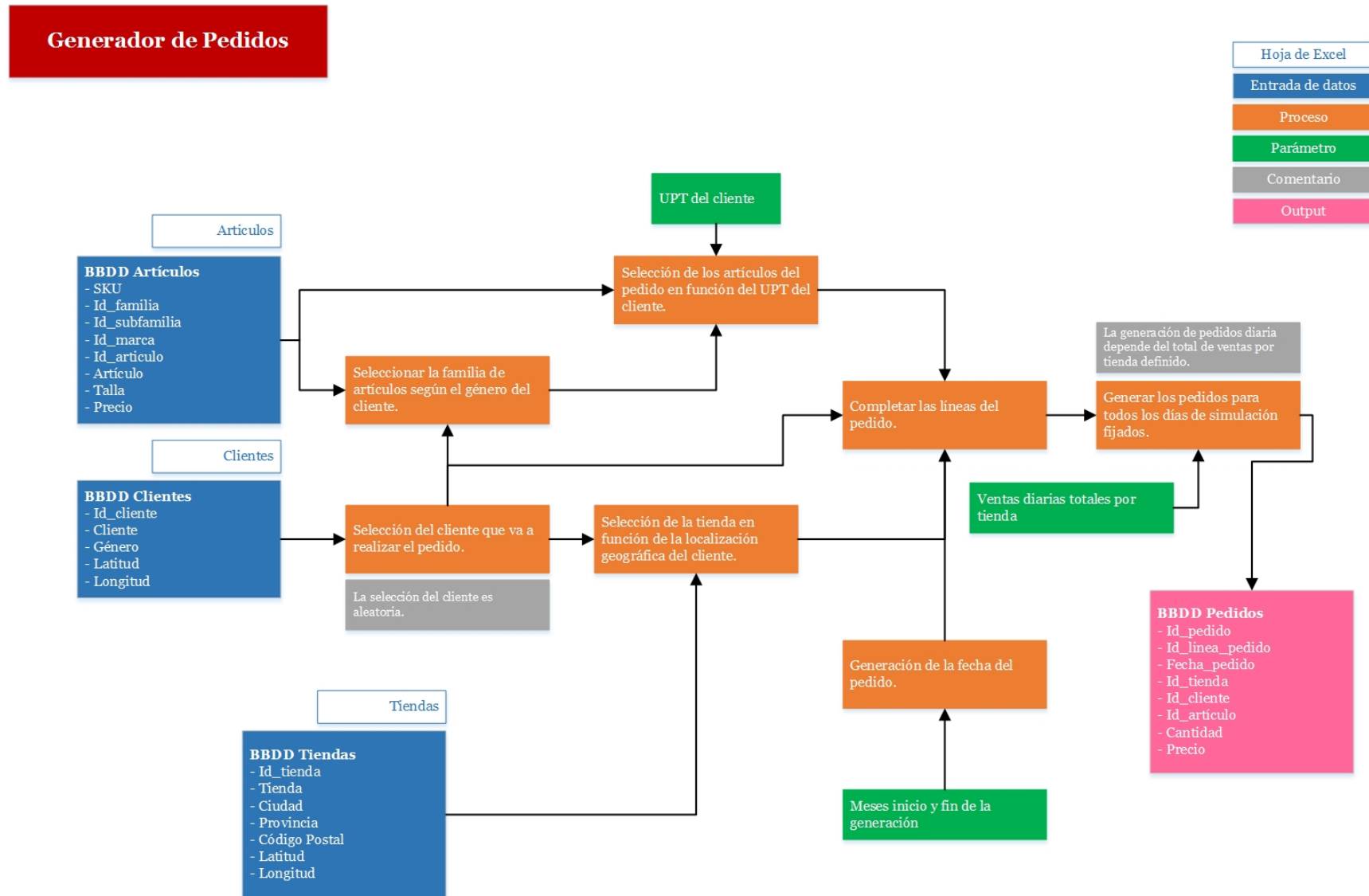


Figura 20. Diagrama de flujo del generador de pedidos.

7.5. Generador de devoluciones

El generador de devoluciones, al igual que el generador de pedidos, es una herramienta complementaria al OMS. Éste ha sido programado siguiendo una lógica similar a la del generador de devoluciones. A través de un código de programación realizado con el lenguaje Python, se emplea una lógica secuencial aleatoria que toma como inputs la tabla generada con el generador de pedidos. Además, como input adicional toma de la tabla de Tiendas los valores definidos en la columna “Porcentaje de devoluciones”, que determina el porcentaje de los pedidos recibidos que serán devueltos por los clientes correspondientes.

El generador toma aleatoriamente los ID de los pedidos y los selecciona como devueltos, creando una tabla como la creada en el generador de pedidos donde se van añadiendo las líneas de devoluciones con su correspondiente codificación. Además, se completan datos como la fecha de devolución, que ha sido establecida siguiendo la regla de que los clientes devuelven el artículo adquirido al día siguiente. Es cierto que este hecho puede alejarse ligeramente de la realidad pero era la manera más sencilla de establecer la lógica de programación correspondiente.

El output del generador también es una tabla que forma parte de la base de datos del sistema y que guarda la integración propia de la base de datos relacional a través de la tabla de Pedidos. El formato de dicha tabla es muy similar al de la tabla de Pedidos salvo que incluye la fecha de la devolución. A continuación, en la Tabla 9, se muestra un extracto de dicha tabla debido a que, al igual que la obtenida con el generador de pedidos, el tamaño de la misma es demasiado extenso como para ser mostrada al completo.

ID Devolución	ID Pedido	Fecha Pedido	Fecha Devolución	ID Tienda	ID Cliente	ID Artículo	Cantidad	Precio
1	153	2017-10-01 00:10:12	2017-10-02 00:10:12	3	81	207	1	54,50 €
2	124	2017-10-01 00:19:26	2017-10-02 00:19:26	15	99	648	1	249,50 €
3	5	2017-10-01 01:38:39	2017-10-02 01:38:39	13	13	34	1	250,00 €
4	15	2017-10-01 02:00:00	2017-10-02 02:00:00	6	56	348	1	90,00 €
5	31	2017-10-01 02:42:13	2017-10-02 02:42:13	15	4	636	1	27,95 €
...	...	...	...	...	...	...	...	...
2182	13826	2017-12-30 22:34:04	2017-12-31 22:34:04	3	67	41	1	230,00 €

Tabla 9. Extracto de la tabla de devoluciones.

A modo de resumen de lo obtenido con la generación de los pedidos y las devoluciones para cada una de las tiendas, se ha realizado una gráfica en la que se recopilan los valores de ambos flujos para cada una de las tiendas y el almacén.

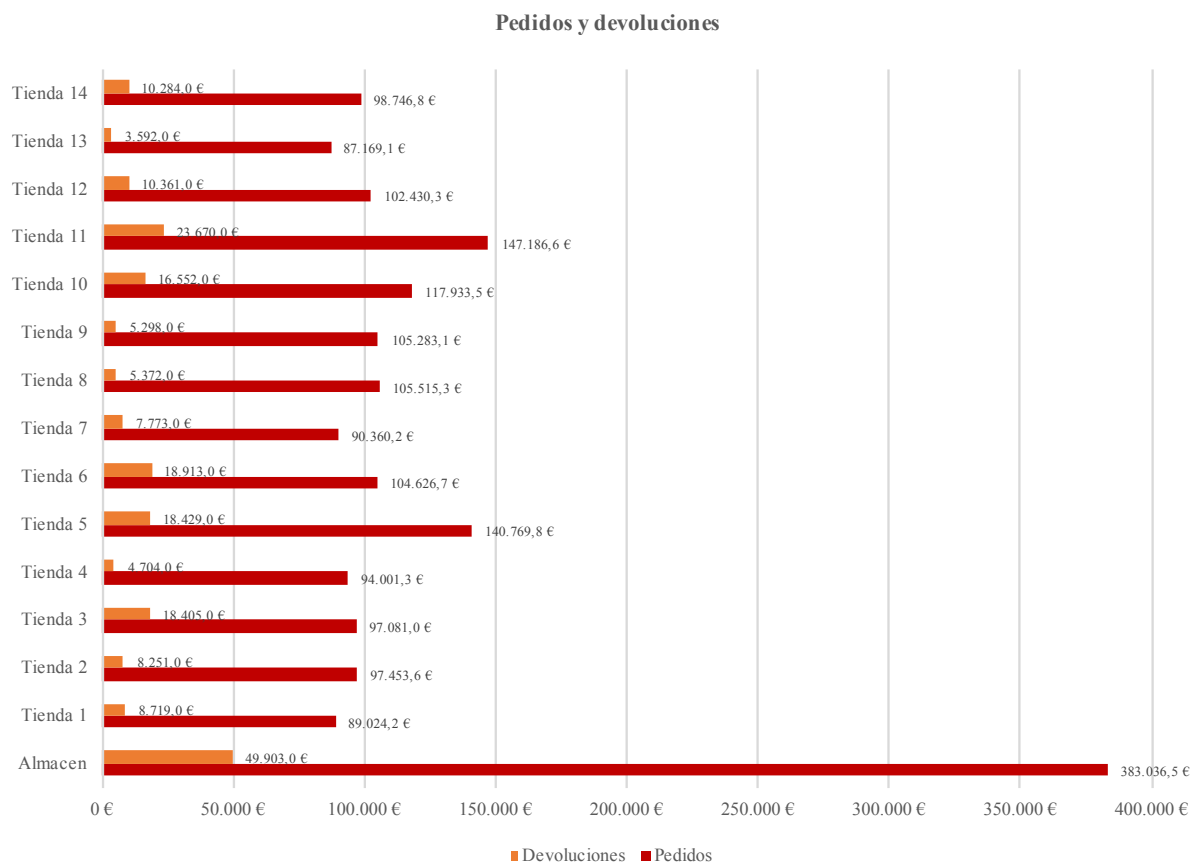


Figura 21. Pedidos y devoluciones generadas para cada una de las tiendas.

Tal y como se definió, las tiendas de Barcelona y de Madrid definidas como tiendas *flagship* son aquellas que mayor impacto económico tienen en el negocio. Además, también se había definido que el almacén tuviese tres veces más pedidos que una tienda normal para poder, de esta forma, cuantificar los pedidos online y su impacto sobre el modelo.

Al igual que con el generador de pedidos, se muestra en la Figura 22 el diagrama de flujos de la programación del generador en cuestión para facilitar la comprensión del mismo de forma visual. Por otro lado, el código de programación se encuentra en el Anexo 2: Generador de Devoluciones.

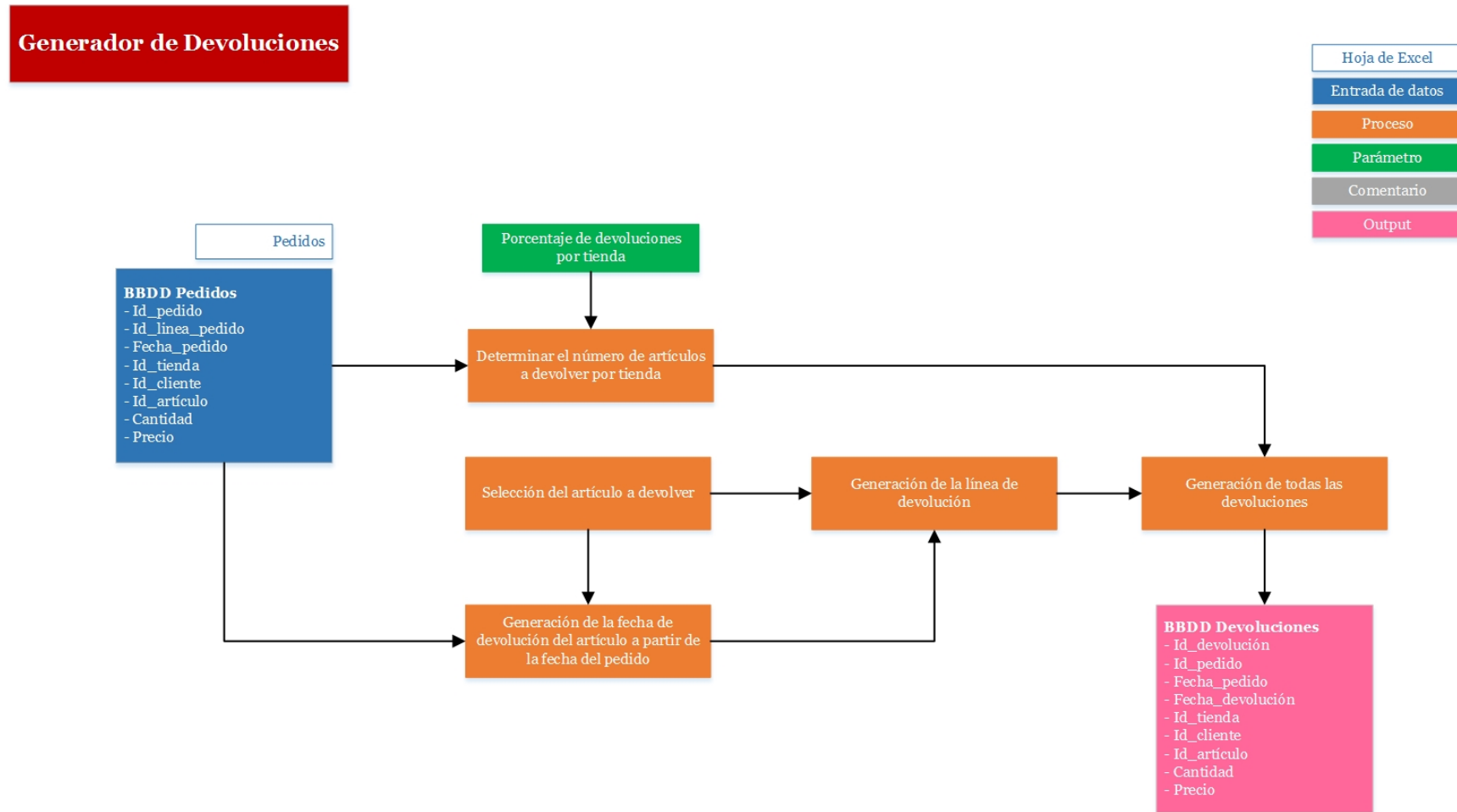


Figura 22. Diagrama de flujo del generador de devoluciones.

7.6. Generador de stock inicial

El generador de stock inicial también es una herramienta clave en el desarrollo del modelo de gestión de pedidos debido a que el output de dicho generador será uno de los inputs del OMS. El generador de pedidos está formado por tres secciones, dentro de las cuales se generan tres tablas como elementos de salida.

1. Generación del stock ideal.
2. Generación del stock inicial.
3. Redistribución del stock inicial.

En la primera sección, la lógica de programación establecida consiste en la comparación de los dos inputs que recibe el generador, que son la tabla de pedidos y la de artículos. El procedimiento es el de recorrer la tabla de pedidos e ir completando la tabla de inventario tienda por tienda, haciendo recuento de los artículos que han sido solicitados. Por tanto, el resultado será una tabla que contenga todos los artículos existentes en el catálogo de la compañía definida en el proceso con los niveles de stock establecidos.

Con la generación de stock ideal lo que se pretende es poder hacer una comparación de los resultados del modelo con lo que sería el caso ideal de estudio, en el cual los niveles de stock cubrirían a la perfección la demanda, sin excesos ni defectos de stock.

La segunda sección de este generador lo que realiza es la modificación del stock ideal en base a un variable en la que se indica el porcentaje de defecto o exceso de stock que debe tener cada tienda. Esta variable es un input del generador y se encuentra en la tabla de Tienda de la base de datos. Teniendo en cuenta esto, se muestra en la Figura 23 dichos niveles de stock definidos.

Por tanto, tomando el porcentaje definido para cada tienda, el generador analiza el stock ideal de cada una de las tiendas y de manera aleatoria modifica los niveles de stock en función del valor establecido en la variable de entrada.

Lo que se pretende con esto es que el modelo de gestión de pedidos tenga un cierto nivel de realismo y que, con ello, se puedan establecer diferentes lógicas de resolución de la distribución de los pedidos en la red de distribución. Además, también se pretende con esto añadir diferentes costes que permitan penalizar el escenario planteado en el caso de la ausencia de un artículo en concreto en stock.

Por último, la tercera sección de del generador tiene como objetivo llevar a cabo una redistribución del stock obtenido en la sección 2 para el almacén a través de las tiendas de la red logística. El motivo de realizar esta redistribución es para dejar el nivel de inventario del almacén al mínimo y que en los escenarios en los que los pedidos de la tienda online se sirvan del stock de las tiendas haya mayor dinamismo y menos acumulación de stock en el almacén. El resultado de las dos últimas secciones del generador de stock se muestra en la Figura 24,

donde se indican los niveles de stock por tienda antes y después de la redistribución llevada a cabo.

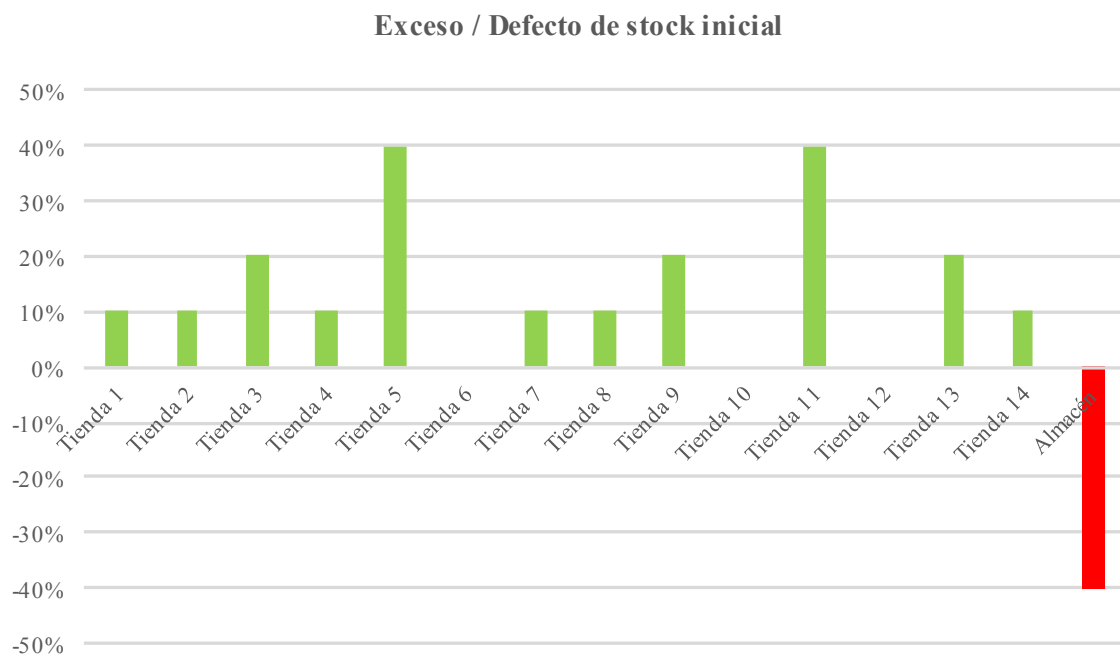


Figura 23. Exceso / Defecto de stock inicial por tienda.

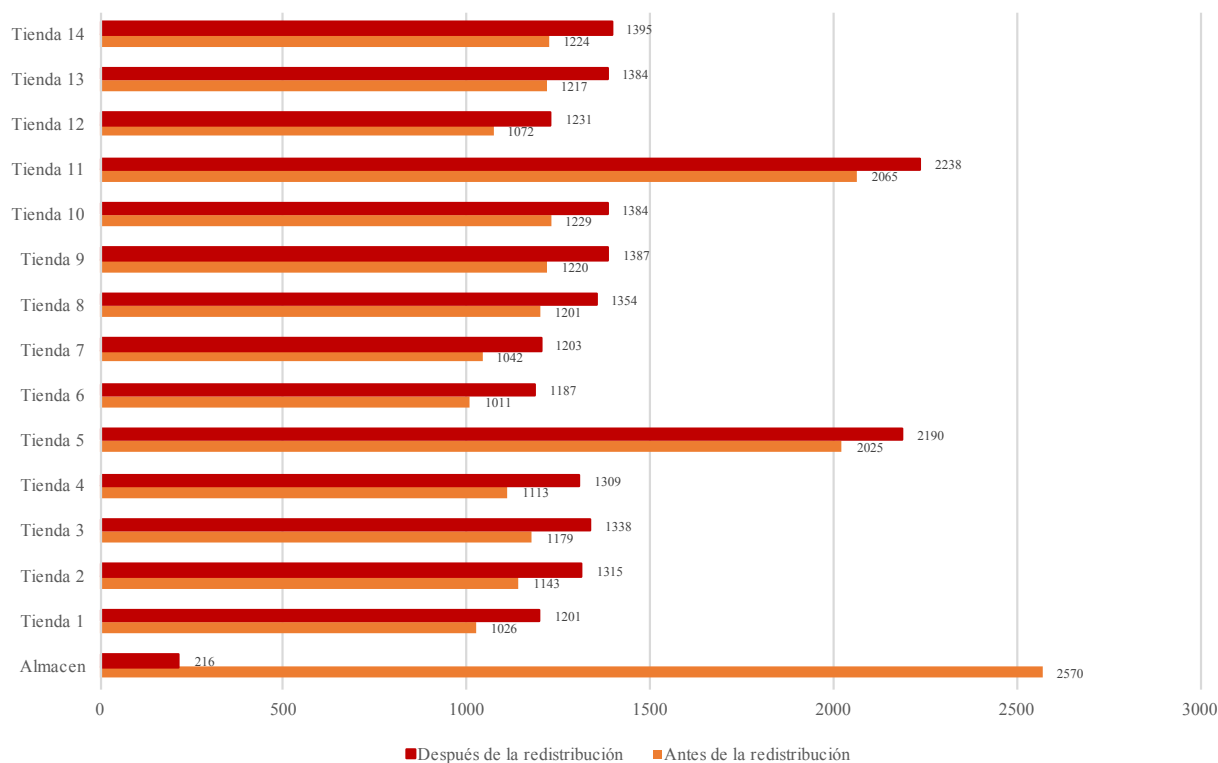


Figura 24. Niveles de stock obtenidos con el generador de stock.

Como resultado de estas tres secciones se obtienen tres tablas que serán empleadas en cada uno de los escenarios correspondiente que se van a plantear en el apartado siguiente. Las tres variantes del inventario han sido generadas de acuerdo con los criterios establecidos para la configuración del OMS. Todo esto se planteará en el apartado siguiente.

El formato de las tres tablas de salida del generador es el mismo. En resumen, consta de cinco columnas en las cuales se presentan la codificación de la tienda correspondiente y la de cada una de los artículos que conforman el catálogo. Además, hay otras tres columnas que muestran la cantidad de unidades que hay por cada artículo en cada tienda seguido del precio unitario de cada artículo y, finalmente, el valor total de las unidades existentes de cada artículo en cada una de las tiendas. El formato explicado se muestra en la Tabla 10, en la cual se muestra un extracto de una de las tablas generadas.

ID Tienda	ID Artículo	Cantidad	Precio	Cantidad Monto
1	1	2	34,96 €	69,92 €
1	2	3	34,96 €	104,88 €
1	3	3	34,96 €	104,88 €
1	4	0	34,96 €	- €
1	5	6	169,00 €	1.014,00 €
1	6	2	169,00 €	338,00 €
1	7	2	169,00 €	338,00 €
1	8	2	169,00 €	338,00 €
1	9	6	175,00 €	1.050,00 €
1	10	3	175,00 €	525,00 €
...	...	...	...	...
15	778	8	49,95 €	399,60 €
15	779	0	49,95 €	- €

Tabla 10. Extracto de la tabla de stock inicial.

Al igual que en los dos generados presentados anteriormente, el código de programación se adjunta en el Anexo 3: Generador de Stock Inicial. Por otra parte, se muestra en la Figura 25 el diagrama de flujo correspondiente, en el cual se muestra la lógica llevada a cabo en la programación del generador.

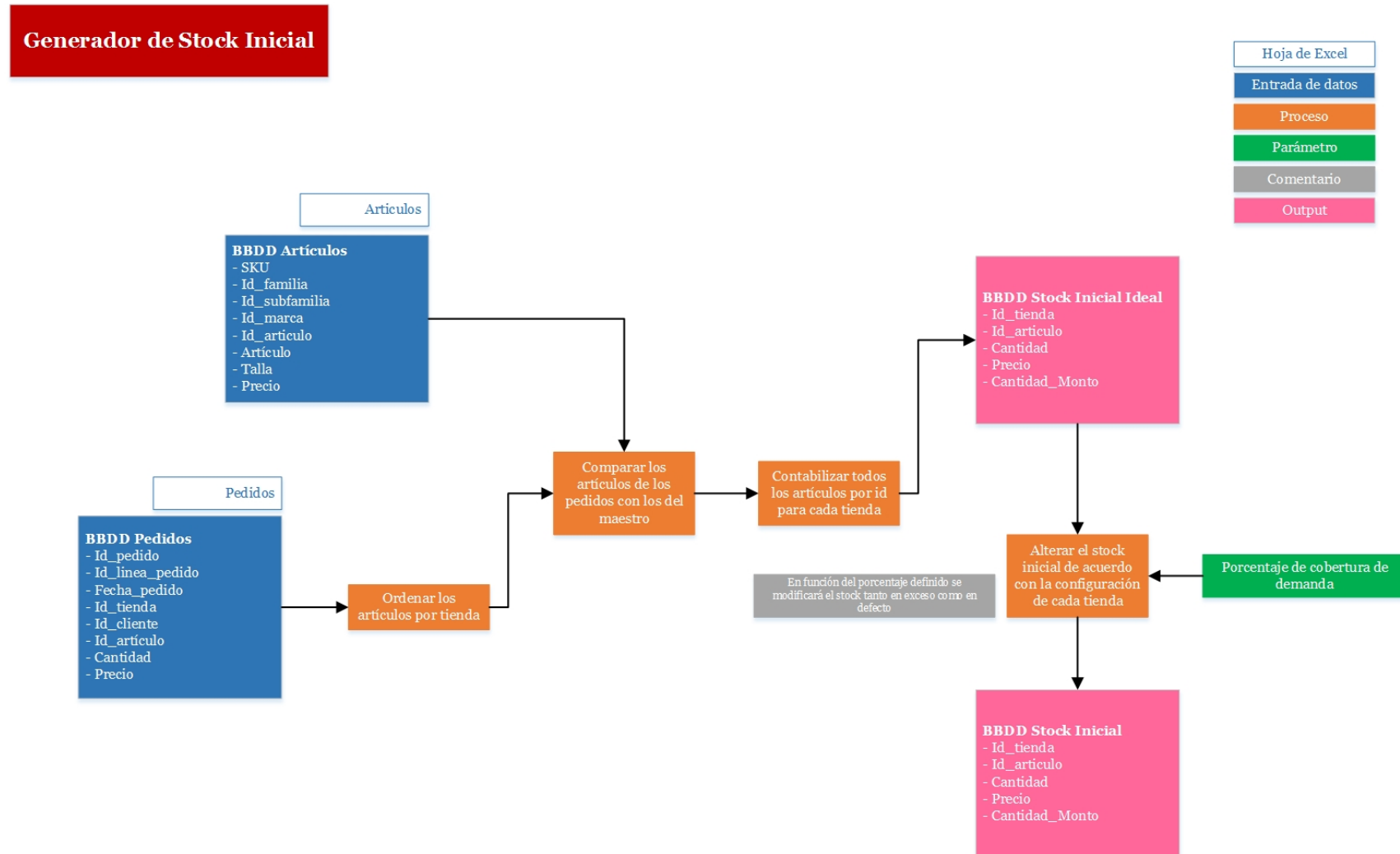


Figura 25. Diagrama de flujo del generador de stock inicial.

7.7. OMS

Un sistema de gestión de pedidos (*Order Management System, OMS*) proporciona un lugar centralizado para gestionar los pedidos procedentes de todos los canales de venta. Centralizar esto en un solo sistema es fundamental para brindar una experiencia de cliente superior al proporcionar el estado del pedido, las entregas a tiempo y cumplir con las expectativas del cliente para comprar, cumplir y devolver en cualquier lugar.

La solución adecuada le permite disponer de una base para una ejecución uniforme y sin problemas en todos los aspectos relacionados con clientes, finanzas, inventario y cadena de suministro que impulsan la satisfacción de los consumidores, la repetición de negocios y la lealtad a largo plazo. Además, los sistemas de administración de pedidos pueden mejorar la visibilidad de las ventas y reducir las demoras y los pedidos pendientes al brindarle una vista única de la demanda, el inventario y el suministro.

Un sistema de gestión de pedidos eficaz le permite organizar sin problemas el cumplimiento de los productos y servicios en prácticamente todos los puntos de contacto de sus clientes. Sin un sistema centralizado de administración de pedidos, lograr una visión integral de la demanda, las transacciones y los clientes en todos los canales y ubicaciones de cumplimiento puede ser extremadamente difícil.

La gestión de pedidos le permite pensar más allá del inventario disponible para centrarse en lo que está disponible para prometerle a sus clientes. Elimina la costosa y rígida segmentación de inventario en todos los canales y unidades de negocios con una visión única y en tiempo real del inventario en toda la empresa, desde sus proveedores y plantas de fabricación hasta su almacén y tiendas.

Con una vista unificada, puede realizar el ideal de poner a disposición su inventario completo en una posición vendible para cualquier cliente a través de cualquier canal. Puede fomentar la lealtad de los clientes al ofrecer la experiencia rápida, precisa e ininterrumpida que exigen los clientes mientras mejora su rentabilidad operativa.

Como punto adicional a señalar, el OMS permite realizar una mejora de la estrategia logística, analizar la red de distribución, ver el impacto económico de cada estrategia y, además, permite optimizar la apertura de nuevos centros logísticos (almacenes/tiendas) en la red de distribución.

7.7.1. Escenarios y criterios de diseño

El objetivo fundamental de la elaboración del OMS objeto de este proyecto es tener la posibilidad de estudiar distintos escenarios que podrían darse en el día a día de las empresas del sector retail. Estos escenarios se corresponden con estrategias que las compañías podrían desarrollar con el fin de optimizar su red de distribución.

La ejecución de estos escenarios se ha llevado a cabo desarrollando modificaciones sobre el código de programación base del OMS, añadiendo las características propias de cada uno de ellos. Por otro lado, los datos de entrada al OMS serán aquellos obtenidos por los generadores construidos, explicados en los apartados anteriores.

Fundamentalmente, el OMS recibirá la base de datos de pedidos, en la cual se recogen cada uno de los pedidos que la empresa recibe a lo largo de los días comprendidos en el período de simulación. Además, para la correcta ejecución del OMS en la mayoría de los escenarios que se explicarán a continuación, es necesaria la elaboración de una matriz de cada uno de los puntos de la red de distribución donde se recojan las distancias que hay entre cada uno de los puntos respectivamente.

Como resultado del modelo, para cada escenario se obtendrán dos tablas de salida. En primer lugar, una tabla en la cual se presentará la asignación de cada uno de los pedidos, es decir, de qué tienda origen se tomará el artículo del inventario seleccionado para poder abastecer la demanda. En segundo lugar, se obtendrá una tabla con una serie de indicadores tanto de costes como de actuación que serán de gran utilidad para poder analizar el comportamiento del modelo en cada uno de los escenarios simulados.

En cuanto a las variables de salida, la tabla que las contiene muestra un desglose de una serie de variables tanto para la venta física como para la online, mostrando las variables para cada una respectivamente. Con estas variables se podrán desarrollar posteriormente los informes de cada uno de los escenarios con la ayuda de la herramienta Power BI. Los costes mencionados se muestran a continuación:

- Coste de Transporte
- Coste de Empaquetado
- Coste de Mano de Obra
- Coste de Rotura de Stock
- Coste total
- Número de pedidos no servidos

A continuación, se muestra una tabla en la que se explican brevemente los aspectos comunes de cada uno de los escenarios y los criterios empleados para caracterizar cada uno de ellos.

<i>Escenarios</i>	<i>Venta física</i>	<i>Pedidos online</i>	<i>Stock Inicial tiendas</i>	<i>Stock inicial almacén</i>	<i>Stock de seguridad</i>	<i>Rotura stock</i>	<i>Costes</i>	<i>Outputs</i>
Escenario 1	Las compras en tienda física se abastecen con el stock de las tiendas	Los pedidos online se abastecen con el stock de los almacenes	Parametrizado en el maestro de tiendas con exceso o defecto	El stock de los pedidos online se surte de un solo almacén que tiene la logística incorporada para ello	No	Todos los pedidos que en tienda u online no puedan ser abastecidos, no será posible surtirlos y se contabilizará como venta perdida	Coste de transporte, coste packaging, coste mano de obra, coste de rotura de stock	Tabla de asignación de pedidos, tabla con datos de costes y rotura de stock
Escenario 2	Las compras en tienda física se abastecen con el stock de las tiendas.	Los pedidos online se abastecen con el stock de cualquier tienda o almacén en función de la cercanía y la disponibilidad de stock.	Parametrizado en el maestro de tiendas con exceso o defecto	Se reparte entre las tiendas un porcentaje de los pedidos y el otro porcentaje se queda en almacén	No	Todos los pedidos que en tienda u online no puedan ser abastecidos, no será posible surtirlos y se contabilizará como venta perdida	Coste de transporte, coste packaging, coste mano de obra, coste de rotura de stock	Tabla de asignación de pedidos, tabla con datos de costes y rotura de stock
Escenario 3	Las compras en tienda física se abastecen con el stock de las tiendas.	Los pedidos online se abastecen con el stock de cualquier tienda o almacén en función de la cercanía y la disponibilidad de stock.	Parametrizado en el maestro de tiendas con exceso o defecto	Se reparte entre las tiendas un porcentaje de los pedidos y el otro porcentaje se queda en almacén	Stock de seguridad en tiendas. El nivel de stock dependerá del tipo de tienda	Todos los pedidos que en tienda u online no puedan ser abastecidos, no será posible surtirlos y se contabilizará como venta perdida	Coste de transporte, coste packaging, coste mano de obra, coste de rotura de stock	Tabla de asignación de pedidos, tabla con datos de costes y rotura de stock
Escenario 4	Las compras en tienda física se abastecen con el stock de las tiendas. En caso no haber en tienda más cercana, existe la posibilidad de intertienda.	Los pedidos online se abastecen con el stock de cualquier tienda o almacén en función de la cercanía y la disponibilidad de stock.	Parametrizado en el maestro de tiendas con exceso o defecto	Se reparte entre las tiendas un porcentaje de los pedidos y el otro porcentaje se queda en almacén	Stock de seguridad en tiendas. El nivel de stock dependerá del tipo de tienda	Todos los pedidos que en tienda u online no puedan ser abastecidos, no será posible surtirlos y se contabilizará como venta perdida	Coste de transporte, coste packaging, coste mano de obra, coste de rotura de stock	Tabla de asignación de pedidos, tabla con datos de costes y rotura de stock

Tabla 11. Resumen de los criterios de diseño de los escenarios del OMS.

7.7.2. Escenario 1

El primer escenario sobre el cual se aplicará el sistemas de gestión de pedidos construido está basado en una configuración de la red de distribución que actualmente es muy común en las empresas, sobre todo, pequeñas y medianas del sector. Dicha configuración permite establecer un sistema centralizado de gestión de los pedidos de la tienda online, almacenando el inventario de la misma en un único almacén de la red logística. En el caso de la red en estudio solamente hay un único almacén, sin embargo en estructuras mayores puede haber más de uno. Lo importante de esta configuración es el concepto de centralización y de disgregación de la tienda física y la online.

Por lo tanto, tal y como se ha explicado en el párrafo anterior, la división del inventario de la compañía se lleva a cabo siguiendo la división establecida a lo largo de toda la red logística. Las tiendas físicas tienen su propio inventario para cubrir la demanda predicha y el almacén el suyo correspondiente.

Como se comentó en el Apartado 7.6: Generador de stock inicial, el stock inicial de todos los centros se parametrizó de tal forma que se pudiese alterar el nivel con respecto al del stock ideal fijado por la demanda de los pedidos generados con el generador desarrollado. Con objeto de darle dinamismo al modelo y que se pudiese caracterizar adecuadamente la rotura de stock, se le asignó un defecto de stock al almacén mientras que dicho defecto fue corregido con un exceso en algunas de las tiendas restantes para, de esta forma, poder compensar las unidades eliminadas. En la Figura 23, expuesta en apartados anteriores, se muestran los niveles de stock de cada una de las tiendas.

Es cierto que que haya un déficit tan grande en el inventario de un almacén es una situación que, en cierta medida, se aleja de la realidad ya que no es normal que una previsión por parte de una compañía se encuentre tan lejana a los niveles de demanda. Como se ha comentado anteriormente, esta situación ha sido forzada para que el modelo tenga un cierto dinamismo y que los resultados sean más claros debido a que el número de pedidos mensuales no es comparable con el que tiene una tienda del sector. La limitación existente reside en la capacidad de computación a la hora de generar una gran cantidad de datos y tratarlos.

En este escenario, el único factor que influye a la hora establecer las estrategias de distribución es la disponibilidad del artículo correspondiente en el inventario de la tienda en la que se solicite. Por lo tanto, es de esperar, viendo la Figura 23, que la plataforma *e-commerce* será la que sufra una mayor cantidad de roturas de stock debido a que es la única que tiene un déficit de unidades en el inventario inicial.

La lógica de resolución del OMS para este escenario se explica de forma esquemática en la Figura 26, en la cual se muestra un diagrama de flujo del funcionamiento del modelo.

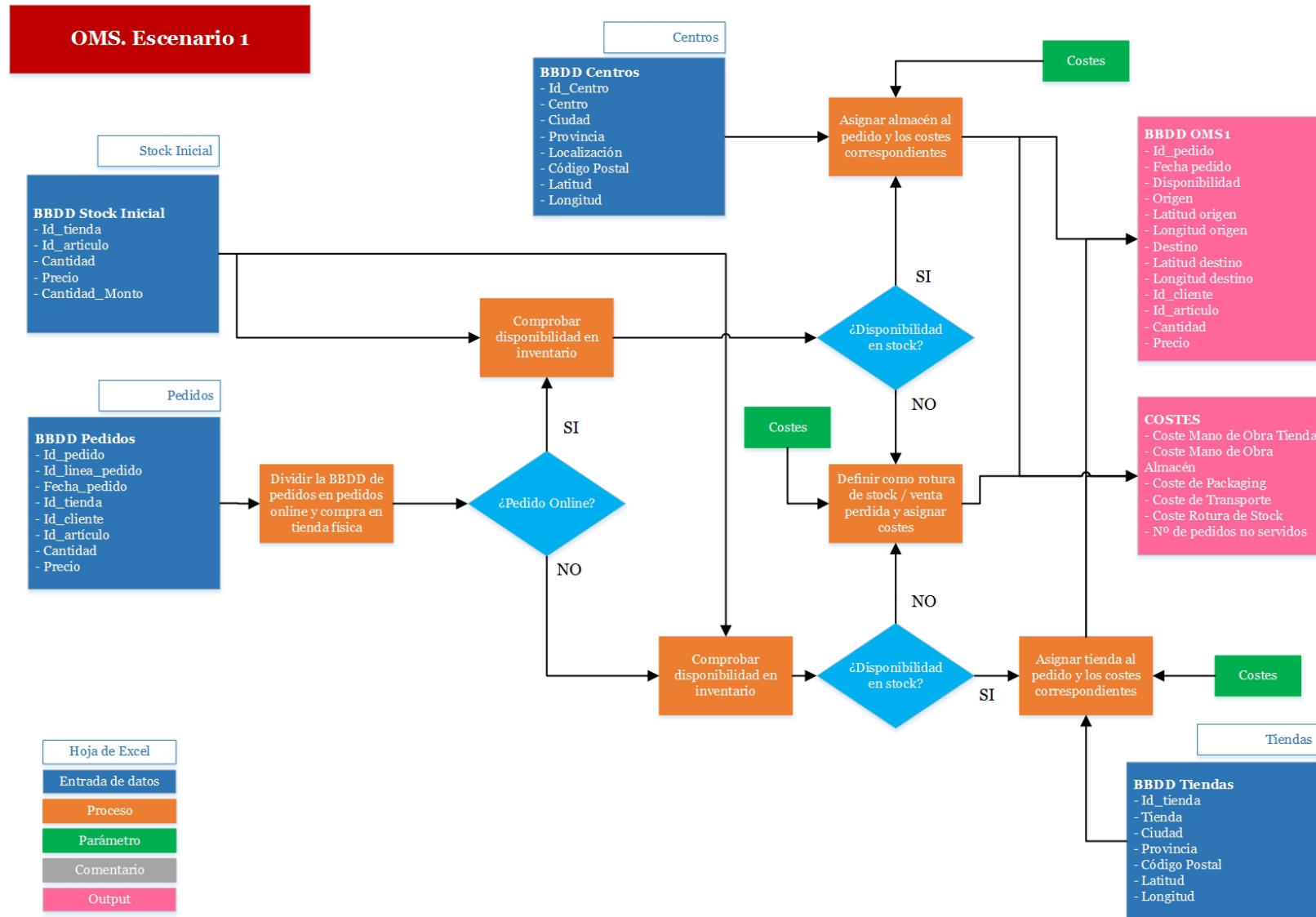


Figura 26. Diagrama de flujo del OMS. Escenario 1.

El funcionamiento del OMS para este escenario parte de cada una de las líneas de pedido de la tabla de la base de datos generada con el generador. Teniendo en cuenta el identificador de la tienda de dicha tabla, se determina si el pedido es proveniente de la tienda online o se corresponde con una compra de la tienda física. El objetivo de esta distinción se debe a que la información de las tiendas y los centros o almacenes se encuentran en tablas distintas dentro de la base de datos.

El siguiente paso sería comprobar la disponibilidad del artículo correspondiente al de la línea de pedido en curso comparando con la tabla de inventario de la tienda o almacén según corresponda. En caso favorable, se asigna el destino dentro de la tabla de salida del modelo y los correspondientes costes de transporte, mano de obra, etc.

El coste de transporte se determina haciendo uso de una matriz creada en la cual se muestran las distancias entre cada uno de los puntos de la red de distribución, incluyendo los domicilios de los clientes que llevan a cabo compras por internet. Tomando un coste por kilómetro, definido anteriormente, se establece el coste de transporte de dicha operación.

En el caso en el que no haya disponibilidad de dicho artículo en stock, se indicará en la tabla de salida marcando con un 0 la columna de “Disponibilidad”. Además, se imputará la venta perdida con un coste que se corresponde con el coste medio de los artículos solicitados y se incrementará la cuenta de la rotura de stock.

La venta perdida se contabilizará tanto para la tienda física como para la venta online. Cabe destacar que esta acción no siempre puede ser contabilizada debido a que los negocios no pueden determinar cada vez que un cliente solicite un artículo y no haya en stock. Sin embargo, con los sistemas de consulta instalados en las tiendas de última generación se pueden estudiar las búsquedas que los clientes hacen y a través de las mismas hacer estimaciones de la rotura de stock. En consecuencia, la estimación de este número resulta ser un indicador que las compañías emplean para mejorar sus previsiones de demanda y estudiar los posibles patrones y, además, corregir errores cometidos. Esta metodología se empleará en los siguientes escenarios también.

El resultado de la simulación del escenario en estudio mediante el sistema de gestión de pedidos da como resultados la tabla de salida de asignación de pedidos. Debido a las dimensiones de la misma, se muestra a continuación una matriz en la que se muestra la asignación de los pedidos totales entre origen y destino con el objetivo de poder analizar el comportamiento del modelo. Esta matriz se muestra en la Tabla 12, siendo los centros origen las filas y los destinos mostrados en las columnas.

ORIGEN	Domicilio cliente	Tienda 1	Tienda 2	Tienda 3	Tienda 4	Tienda 5	Tienda 6	Tienda 7	Tienda 8	Tienda 9	Tienda 10	Tienda 11	Tienda 12	Tienda 13	Tienda 14	Total
Almacén	3917															3917
Tienda 1		932														932
Tienda 2			1039													1039
Tienda 3				983												983
Tienda 4					1012											1012
Tienda 5						1447										1447
Tienda 6							1011									1011
Tienda 7								947								947
Tienda 8									1092							1092
Tienda 9										1017						1017
Tienda 10											1229					1229
Tienda 11												1476				1476
Tienda 12													1073			1073
Tienda 13														1015		1015
Tienda 14															1113	1113
Total	3917	932	1039	983	1012	1447	1011	947	1092	1017	1229	1476	1073	1015	1113	19303

Tabla 12. Distribución de pedidos origen/destino OMS. Escenario 1.

Observando los resultados de la Tabla 12, se ve claramente que el modelo se comporta según lo esperado ya que la asignación de los pedidos se ha ejecutado cumpliendo el criterio impuesto. Es decir, los pedidos de la tienda online se abastecen con el inventario del almacén, mientras que las tiendas se encargan de cubrir la demanda existente en la venta física.

En cuanto a los indicadores obtenidos del estudio de la solución del OMS para este escenario, se muestran a continuación.

Unidades servidas en tienda origen	15386
Unidades servidas en otra tienda (intertienda)	0
Pedidos servidos en tienda	0
Pedidos servidos en almacén	3917
Unidades servidas totales	19303
Precio medio unidad servida	94,40 €
Venta en tienda no servida (OMS)	0
Venta en tienda no servida (estimación)	1449
Pedidos online no servidos	369
Unidades totales no servidas	1818
Rotura de stock	9,24%

Para este escenario, los indicadores más significativos son los relacionados con la venta perdida. En primer lugar, tal y como se comentó anteriormente, la venta perdida tendrá lugar en el sistema de venta *e-commerce* debido a que el inventario del almacén es el único que se encuentra con un nivel negativo de unidades y, según los criterios de diseño del escenario en cuestión, no se puede abastecer dicha demanda con las unidades de otro inventario.

Teniendo en cuenta el número de pedidos online no servidos, se puede hacer una estimación de cuál sería la rotura de stock en tienda mediante una relación de proporcionalidad con el número total de unidades servidas en cada uno de los sistemas de venta existentes. Por lo tanto, habrá una relación 1 : 4 aproximadamente de unidades no servidas entre la venta física y la online. Todo esto se recoge en un 9,24% de rotura de stock en este escenario, valor que según los criterios mostrados en la Tabla 11 debería ser el más alto debido a la poca flexibilidad que tiene el modelo.

Finalmente, el código de Python del modelo se adjunta en el Anexo 4: OMS. Escenario 1.

7.7.3. Escenario 2

Este nuevo escenario parte de la configuración de la red de distribución definida en el escenario anterior. Sin embargo, se incluye una nueva variante en la configuración de la misma que permite un aumento de la flexibilidad del modelo y, como consecuencia, mayor dinamismo y eficiencia.

El sistema de abastecimiento de la demanda en tienda física se mantiene tal cual fue definido en el escenario anterior, es decir, cada tienda tiene su inventario correspondiente con el cual debe de cubrir la venta. Por el contrario, la gestión de los pedidos provenientes de la plataforma online se ve modificada debido a que ahora los pedidos pueden ser servidos desde el inventario tanto del almacén de la red logística como desde el del resto de tiendas.

En relación con el stock inicial de este nuevo modelo, los niveles de defecto y exceso de stock se mantienen según lo definido en la Figura 23. El inventario del almacén sigue siendo el único que se encuentra en niveles negativos. Sin embargo, para este escenario la distribución de las unidades que conforman el inventario total ha sido alterada con respecto a la configuración anterior.

Según lo mostrado en la Figura 24, el inventario del almacén se ha reducido considerablemente con respecto al del caso anterior debido a que el modelo ha sido parametrizado con un porcentaje de reparto de dicho inventario entre el resto. El motivo fundamental de esta decisión se debe a que las probabilidades de sufrir acumulaciones de stock en un almacén son bastante más altas que las de sufrirlas en una tienda. Las unidades de inventario de una tienda pueden servir para abastecer, en este caso, la venta física y la venta online. Básicamente, las acumulaciones de stock implican costes adicionales para la compañía, que con una gestión eficiente pueden ser reducidos considerablemente.

Para el escenario en estudio, los criterios establecidos como estrategia de distribución de los pedidos son, en primer lugar, la distancia entre el destino del pedido y el origen seleccionado como centro de abastecimiento y, además, la disponibilidad del artículo solicitado en el inventario de la tienda o almacén origen. A diferencia del escenario anterior, se añade la variable de la distancia que, en este caso, es crucial debido al ahorro de costes de transporte que puede suponer.

Como se ha comentado anteriormente, la distancia entre los destinos de los pedidos y los orígenes de los puntos de abastecimiento viene definida en una matriz que constituye uno de los inputs principales del OMS.

Un factor muy importante en el modelo es la temporalidad de los pedidos, es decir, la incertidumbre existente en las tiendas sobre el orden de los pedidos. En otras palabras, en el negocio no saben si el primer pedido va a ser correspondiente con la venta online o si va a ser una compra física en la tienda. Este hecho puede ser un principal causante de roturas de stock debido a que no se puede controlar. A continuación, cuando se muestren los resultados del modelo, se analizará con mayor detalle el efecto de dicho factor en la gestión de los pedidos.

Por otro lado, se muestra en la Figura 27 el diagrama de flujo de la lógica del modelo .

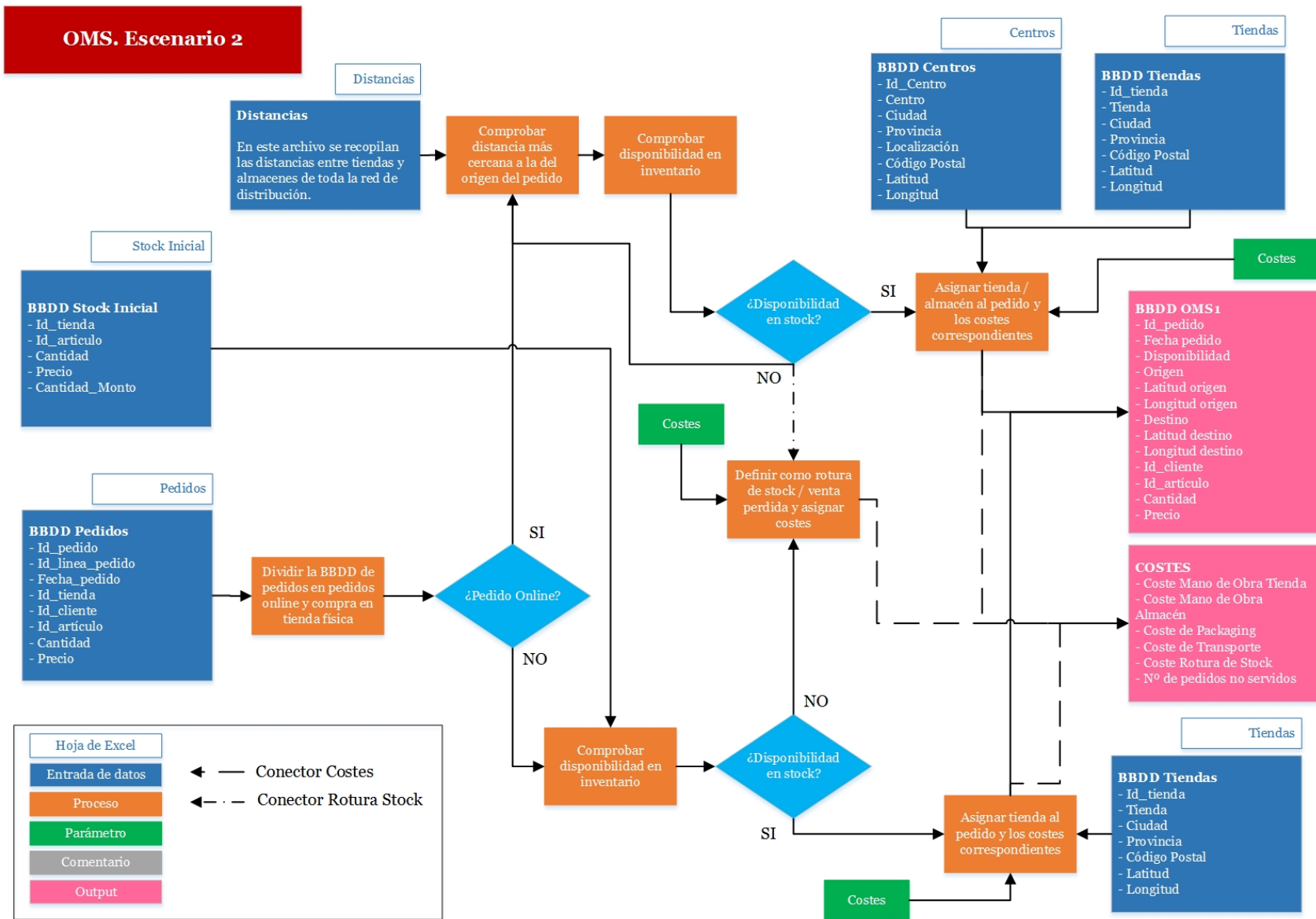


Figura 27. Diagrama de flujo del OMS. Escenario 2.

La lógica del modelo, en este caso, comienza de la misma forma que para el Escenario 1 clasificando los pedidos de la tabla de la base de datos en aquellos provenientes de compra en la tienda física o aquellos que han sido llevados a cabo mediante la tienda online.

El procedimiento de distribución y asignación de la compras en tienda física se mantiene tal cual fue definido en el código de programación del caso anterior. En primer lugar, se comprueba la disponibilidad del artículo seleccionado en el inventario de la tienda correspondiente. Si hay stock disponible, se lleva a cabo la asignación de la tienda origen a dicho pedido y, además, se calculan los costes correspondientes de dicha operación, que para el caso de venta física sería simplemente el coste de mano de obra. Finalmente, la asignación del pedido será cumplimentada en la tabla correspondiente de salida con los datos del pedido mostrados en la Figura 27.

Para el caso de los pedidos de la tienda online, el tratamiento a nivel de código es distinto debido a que ahora no es el inventario del almacén la primera opción. El primer paso que se lleva a cabo es analizar en función del destino cuál sería la opción más viable siguiendo el criterio de proximidad. Este paso es iterativo debido a que depende fundamentalmente de la disponibilidad del artículo correspondiente en el stock de la tienda seleccionada. Por lo tanto, el modelo realizará un proceso iterativo para asignar la tienda o almacén origen en función de ambos factores.

Una vez seleccionado el punto de abastecimiento, se procederá a completar dicha línea de pedido en la tabla de salida del OMS y a la correspondiente asignación de los costes. Para el caso de los pedidos online, habrá que añadir, además del coste de mano de obra, los costes de transporte y de empaquetado. El coste de transporte dependerá de la distancia que haya entre los puntos de origen y destino, mientras que el coste de empaquetado dependerá, como se explicó en apartados anteriores, del tamaño del artículo.

En el caso en el que al disponibilidad del artículo no sea posible, para ambos casos se llevará a cabo el mismo procedimiento. Que contabilizará dicha venta perdida según el procedimiento explicado en el Escenario 1 y se asignarán los costes correspondientes teniendo en cuenta el precio medio de los pedidos.

Los resultados de la simulación del modelo para este escenario se muestran de la misma forma que para el escenario anterior, o sea, mediante la matriz de pedidos distribuidos en la cual se relaciona el origen y el destino de cada una de las unidades surtidas. Esta matriz se muestra en la Tabla 13.

En este modelo el resultado se obtiene tal y como se esperaba debido a que la única variante es que para la columna de Domicilio Cliente hay orígenes distintos repartidos entre cada una de las tiendas de la red de distribución. Por ejemplo, la tienda situada en Galicia únicamente sirve un pedido online debido a su situación con respecto a los clientes que solicitan a domicilio.

ORIGEN	Domicilio cliente	Tienda 1	Tienda 2	Tienda 3	Tienda 4	Tienda 5	Tienda 6	Tienda 7	Tienda 8	Tienda 9	Tienda 10	Tienda 11	Tienda 12	Tienda 13	Tienda 14	Total
Almacén	167															167
Tienda 1	1	930														931
Tienda 2	189		1012													1201
Tienda 3	52			978												1030
Tienda 4	322				976											1298
Tienda 5	107					1446										1553
Tienda 6	14						1010									1024
Tienda 7	321							914								1235
Tienda 8	215								1062							1277
Tienda 9	24									1013						1037
Tienda 10	362										1191					1553
Tienda 11	441											1476				1917
Tienda 12	998												967			1965
Tienda 13	548													986		1534
Tienda 14	525														1077	1602
Total	4286	930	1012	978	976	1446	1010	914	1062	1013	1191	1476	967	986	1077	19324

Tabla 13. Distribución de pedidos origen/destino OMS. Escenario 2.

Los indicadores obtenidos para este modelo son los mismos que para el anterior. A continuación, se mostrarán y se procederán a comentar en base a los criterios de diseño establecidos.

Unidades servidas en tienda origen	15038
Unidades servidas en otra tienda (intertienda)	0
Pedidos servidos en tienda	4119
Pedidos servidos en almacén	167
Unidades servidas totales	19324
Precio medio unidad servida	94,67 €
Venta en tienda no servida (OMS)	348
Venta en tienda no servida (estimación)	1449
Pedidos online no servidos	0
Unidades totales no servidas	1797
Rotura de stock	9,13%

Como se ha comentado anteriormente en este mismo apartado, uno de los factores más influyentes en la simulación de este modelo es la temporalidad con la que los pedidos son solicitados. Este hecho será causa fundamental de los valores obtenidos en los indicadores.

Para la simulación en estudio, la generación de pedidos ha dado lugar a que los artículos para los cuales se ha producido venta perdida pertenezcan a compra en tienda física. Por lo tanto, la compra online de dichos artículos se ha producido antes en el tiempo y, por este motivo, las unidades se han agotado previamente a poder cubrir la demanda en la tienda correspondiente. Al ser la generación de pedidos aleatoria, es posible que en otra simulación el resultado de estos indicadores hubiese sido alterado y los pedidos de la tienda online se vieses afectados en vez de los de la tienda física.

En comparación con el escenario anterior, el número total de unidades servidas ha aumentado debido a que la estrategia de distribución es más eficiente y permite mayor flexibilidad, con lo cual se puede ofrecer una mayor cobertura de los pedidos a causa de la distribución de las tiendas dentro del territorio en el que opera la compañía del proyecto en cuestión. Como consecuencia, el porcentaje de rotura de stock es menor que en el caso anterior a causa de los motivos expuestos.

Por último, el código de programación desarrollado para el OMS de este escenario se encuentra en el Anexo 5: OMS. Escenario 2.

7.7.4. Escenario 3

El objetivo principal de desarrollar el tercer escenario del OMS es para tomar una medida sobre los resultados obtenidos en el escenario anterior. La estrategia seleccionada para la distribución de

los pedidos a domicilio penaliza mucho los resultados de la venta en tienda física debido a que los valores de venta perdida pertenecen únicamente a este medio de venta.

Como es un factor que no puede ser controlado, la medida más eficiente que se puede tomar es la de definir un stock de seguridad para cada una de las tiendas con el fin de evitar que los pedidos online dejen los niveles de stock a cero y que no se pueda disponer de unidades en tienda para la venta física.

Cabe destacar que si los resultados de la simulación del escenario hubiesen sido totalmente opuestos, es decir, que la venta perdida afecta plenamente al negocio *e-commerce*, el escenario 3 sería totalmente distinto ya que se establecería un stock de seguridad para poner remedio al problema existente.

En relación con el stock inicial establecido para este escenario, los niveles fijados de exceso y defecto en cada uno de los centros de almacenamiento son los mismos que en los otros dos casos anteriores. Los porcentajes definidos se muestran en la Figura 23.

Por otro lado, en cuanto a la redistribución del stock del almacén entre las tiendas se mantiene igualmente ya que para este escenario la metodología de distribución es la misma que la del caso número 2. El objetivo fundamental es el de ahorrar costes de almacenamiento innecesario de stock en almacén, tratando de optimizar el inventario de dicho centro a lo largo de la red de distribución. La redistribución mencionada se muestra en la Figura 24, donde las gráficas de barras de color rojo muestran las unidades en stock de cada una de las tiendas y el almacén.

Los factores influyentes en la gestión de los pedidos de este escenario son los mismos que en el anterior, teniendo en cuenta que en el caso actual se ha añadido un nuevo criterio de diseño. El establecimiento del stock de seguridad permite corregir la situación dada en la simulación anterior y, además, incorpora un grado a realidad al modelo. El stock de seguridad es una estimación que las compañías del sector realizan con el fin de evitar una posible rotura de stock. Dependiendo de la tienda el nivel de stock de seguridad variará. Por ejemplo, las tiendas definidas como *flagship* tendrán mayor stock de seguridad que una tienda normal.

Que haya stock de seguridad da lugar a un mayor grado de dificultad para cumplir el criterio de cercanía debido a que habrá que iterar más entre los inventarios de las tiendas para poder encontrar una unidad disponible del artículo solicitado. Por lo tanto, con esta nueva caracterización el modelo tiende a ser más completo y realista.

En la Figura 28, se muestra, al igual que en los otros escenarios, el diagrama de flujo del funcionamiento del OMS para el caso en estudio.

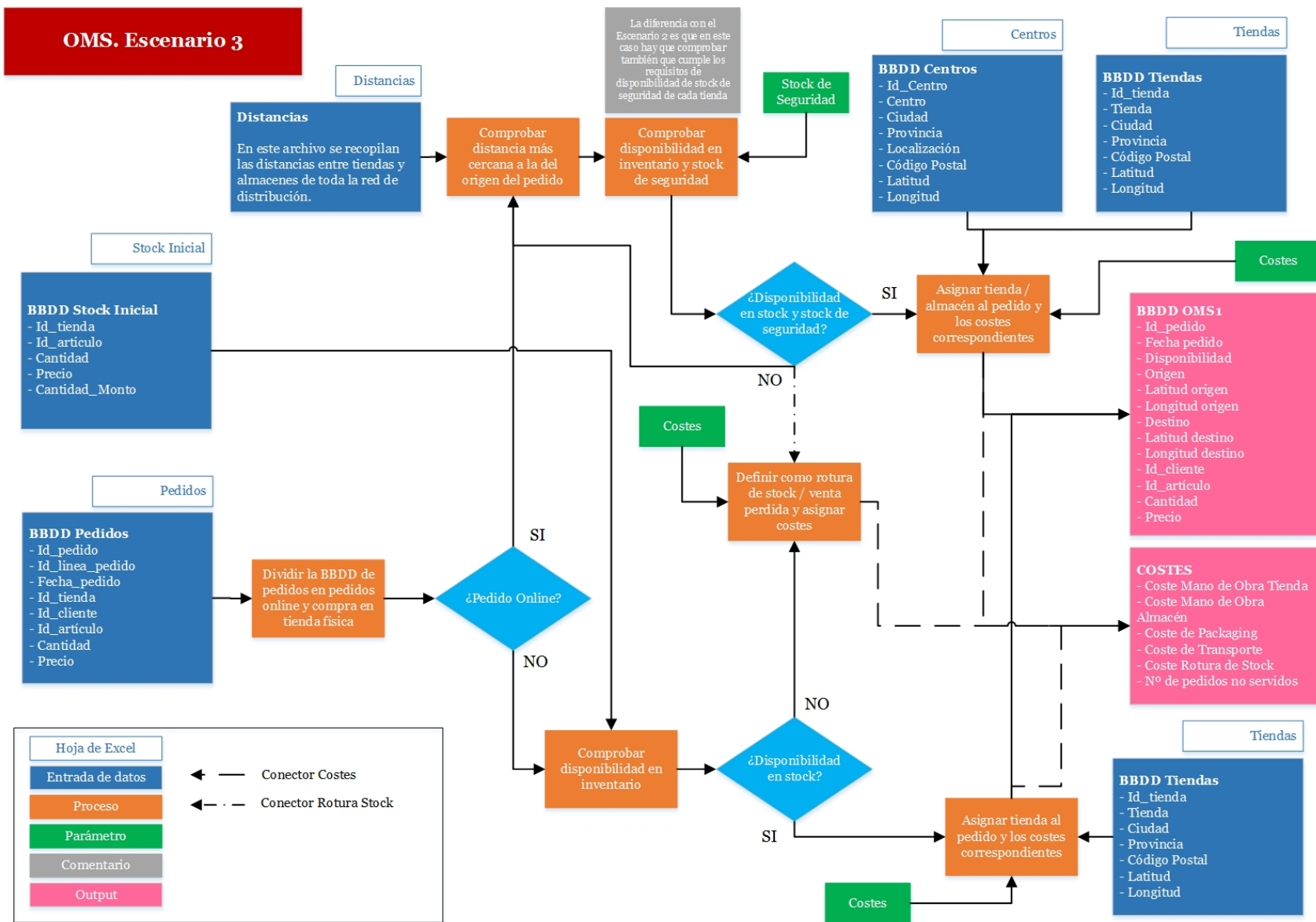


Figura 28. Diagrama de flujo del OMS. Escenario 3.

La lógica de resolución de este modelo es muy similar a la de los anteriores. En primer lugar, se realiza la correspondiente diferenciación entre pedidos online y compra física en tienda mediante el indicador correspondiente de la tienda, perteneciente a la tabla de Pedidos de la base de datos. Una vez diferenciado el tipo de compra, cada tipología tiene un procedimiento distinto a la hora de determinar qué tienda será la que sirva el pedido que se esté distribuyendo en cada momento.

Por un lado, en el caso de la compra física en tienda, la única comprobación que hay que realizar es que haya unidades disponibles en el inventario de la tienda correspondiente del artículo solicitado en el pedido. Para el caso en el que haya disponibilidad, se llevará a cabo la distribución del pedido con la asignación de la tienda origen y los costes correspondientes de dicho proceso. Sin embargo, si no hay unidades suficientes en stock, dicha línea de pedido quedará definida como venta perdida y quedará contabilizada con sus correspondientes costes.

Por otro lado, la distribución de los pedidos online sigue un procedimiento distinto de asignación. En primer lugar, se realizará el proceso iterativo explicado en el apartado anterior con el cual se determinará que tienda/almacén se encuentra más cerca del punto de destino objetivo siguiendo el criterio de cercanía establecido. Este proceso iterativo, realmente, está compuesto por dos pasos de comprobación adicionales, que son los dos criterios restantes definidos para este escenario. Estos pasos son la comprobación de la disponibilidad del artículo solicitado dentro del inventario, en primer lugar, y después que la selección de dicho artículo no implique la ruptura del stock de seguridad definido para dicha tienda. Por lo tanto, es una doble comprobación que permitirá definir si la tienda seleccionada siguiendo el criterio de cercanía puede llegar a servir o no el pedido en curso.

El resultado final de dicho proceso iterativo de selección de tienda origen puede ser bien la selección definitiva de la tienda que cumpla con los criterios establecidos, o bien que no se posible servir dicho pedido y que sea definido como venta perdida. Ambos casos implican una serie de costes correspondientes que se implementarán al modelo en función del resultado obtenido. Finalmente, todos estos datos serán recogidos en la tabla de salida del OMS para este escenario, tanto los datos de la distribución de cada uno de los pedidos, como los costes correspondientes.

En la Tabla 14, se presenta, a modo resumen, los resultados de la asignación de los pedidos para cada una de las tiendas. Aunque los números se presentan de manera global, lo que se pretende con dicha tabla es ver de forma visual el funcionamiento del modelo y, sobre todo, ver que se cumple según lo esperado. Esta tabla junto con los indicadores que se presentarán posteriormente permiten evaluar los resultados sin la necesidad de analizar uno a uno todos los pedidos y su correspondiente distribución.

ORIGEN	Domicilio cliente	Tienda 1	Tienda 2	Tienda 3	Tienda 4	Tienda 5	Tienda 6	Tienda 7	Tienda 8	Tienda 9	Tienda 10	Tienda 11	Tienda 12	Tienda 13	Tienda 14	Total
Almacén	183															183
Tienda 1	16	927														943
Tienda 2	157		1038													1195
Tienda 3	103			973												1076
Tienda 4	270				1012											1282
Tienda 5	80					1447										1527
Tienda 6	15						1010									1025
Tienda 7	353							946								1299
Tienda 8	317								1092							1409
Tienda 9	89									997						1086
Tienda 10	341										1228					1569
Tienda 11	323											1476				1799
Tienda 12	858												1073			1931
Tienda 13	555													1015		1570
Tienda 14	626														1068	1694
Total	4286	927	1038	973	1012	1447	1010	946	1092	997	1228	1476	1073	1015	1068	19588

Tabla 14. Distribución de pedidos origen/destino OMS. Escenario 3.

Comparando las tablas de resultados de los escenarios 2 y 3, se aprecia como hay una variación de los números en ciertas tiendas. En algunas de las tiendas el número de pedidos a domicilio disminuye mientras que el número de compras en tienda aumenta, o viceversa. Este es el efecto del stock de seguridad, que también se ve motivado por los criterios de cercanía y disponibilidad.

En general, la aplicación del stock de seguridad ayuda a mejorar la eficiencia ya que el número total de pedidos servidos aumenta con respecto a los dos casos anteriores. Como se puede ver en los indicadores que se presentan a continuación, el porcentaje de rotura de stock disminuye y, como consecuencia, mejora el comportamiento global del modelo.

Unidades servidas en tienda origen	15302
Unidades servidas en otra tienda (intertienda)	0
Pedidos servidos en tienda	4103
Pedidos servidos en almacén	183
Unidades servidas totales	19588
Precio medio unidad servida	94,60 €
Venta en tienda no servida (OMS)	84
Venta en tienda no servida (estimación)	1449
Pedidos online no servidos	0
Unidades totales no servidas	1533
Rotura de stock	7,79%

Básicamente, la rotura de stock disminuye en un 15% aproximadamente. Además, los pedidos online son todos servidos, con lo que también se mejora la imagen de la compañía y la experiencia que se le provee al consumidor. Aunque es cierto que el número de consumidores que emplea las plataformas *e-commerce* está en constante aumento, la imagen dada en tienda física siempre es muy importante, sobre todo para mantener la fidelidad de los clientes habituales. Por lo tanto, a la vista de los resultados obtenidos, es necesaria la búsqueda de una medida de mejora para disminuir los niveles de venta perdida aún más.

Finalmente, el código de programación del OMS para este escenario se encuentra adjunto en el Anexo 6: OMS. Escenario 3.

7.7.5. Escenario 4

El último escenario del modelo desarrollado para este proyecto cuenta con todo el conjunto de características y criterios que se han ido añadiendo en cada uno de los escenarios anteriores. Además, partiendo de los resultados del Escenario 3, se ha añadido una medida que, en principio, permitirá mejorar el resultado de los números obtenidos en dicho escenario. Por lo tanto, lo que se pretende conseguir es acercar el modelo aún más a la realidad y que el nivel de complejidad sea mayor.

La medida pensada para mejorar el número de pedidos no servidos en tienda es la de añadir un servicio de intertienda. Este servicio permite hacer pedidos a otras tiendas de artículos que no se encuentran disponibles en una tienda concreta. Obviamente, es una opción totalmente voluntaria por parte del cliente y será el mismo cliente el que tendrá que decidir si optar por esa alternativa o no. El objetivo siempre será el de mejorar la experiencia del cliente en el proceso de compra.

En cuanto al stock inicial, para este escenario se emplean los mismos valores que para los dos anteriores. Es decir, el stock del almacén se pretende mantener en niveles mínimos redistribuyendo desde el inicio de la simulación su stock por todas las tiendas de manera equitativa. Esto queda reflejado en la Figura 24.

Los factores que afectan a la simulación de este escenario son principalmente la temporalidad con la que las tiendas reciben tanto los pedidos como las compras y, también, la decisión del cliente en cuanto al servicio intertienda. Es una decisión que se ha modelado con una variable aleatoria, con lo que se podrá acercar más a la realidad el resultado.

Por otro lado, los criterios que se han empleado para caracterizar el OMS son los mismos que el escenario anterior. En primer lugar, el criterio de cercanía, que en este caso será también empleado en el servicio intertienda ya que se ha modelado igual que si fuese un pedido online. El siguiente criterio sería la disponibilidad en el stock de las tiendas que se vayan seleccionando. Por último, el establecimiento de un nivel de stock de seguridad para cada tienda y su correspondiente cumplimiento.

Debido a esto, la capacidad de computación necesaria para simular este modelo será mayor ya que con el criterio de cercanía habrá que hacer dos procesos iterativos para poder encontrar la tienda más cercana que puede servir el pedido en cuestión tanto para la venta online como para el servicio intertienda.

En el diagrama de flujo mostrado en la Figura 29 se puede ver como se representan los procesos para cada uno de los canales de venta con los criterios explicados anteriormente. Al ser el modelo más complejo, se aprecia como el número de condiciones a la hora de asignar una tienda origen a un pedido es mayor.

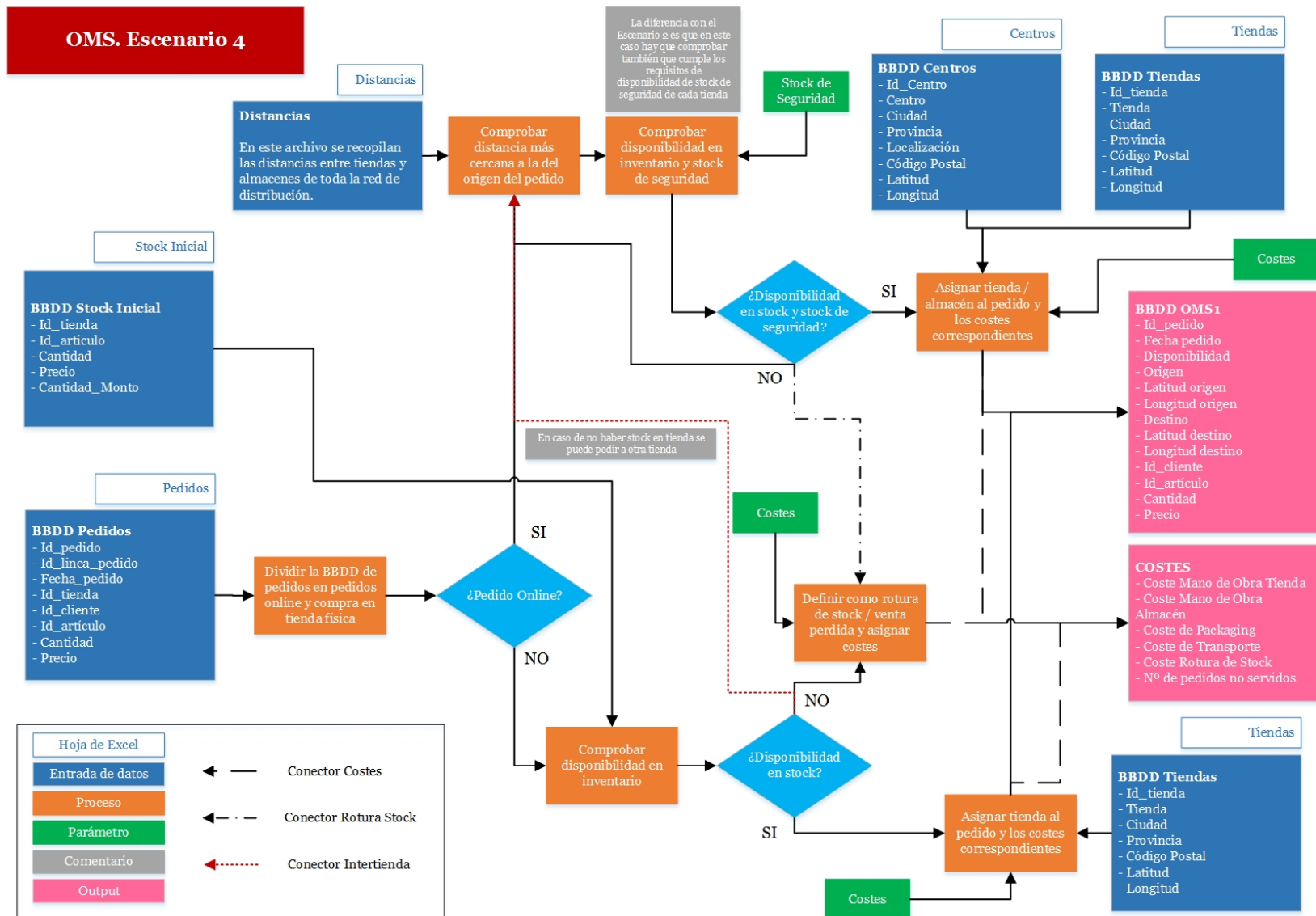


Figura 29. Diagrama de flujo del OMS. Escenario 4.

Con la recepción de un pedido al sistema de gestión (OMS), el primer paso que el modelo lleva a cabo es la diferenciación entre pedido online o compra física en tienda, como se ha realizado en los anteriores usuarios. Esta acción es posible gracias a identificador ID Tienda que forma una de las columnas de la tabla de Pedidos de la base de datos del sistema.

En primer lugar, el procedimiento que se sigue para la asignación de un centro de almacenamiento origen a un pedido comienza comprobando el criterio de cercanía. Haciendo uso de la matriz de distancia se llevará a cabo una comprobación del destino y de los posibles orígenes desde donde se enviará el artículo solicitado. Antes de asignar el origen definitivo habrá que comprobar los otros dos criterios establecidos en el modelo.

Primero, habrá que ver la disponibilidad del artículo que conforma el pedido consultando la tabla de stock de la base de datos. Si está disponible y, además, el stock de seguridad no se ve afectado, la asignación quedará completada. Lo único que habrá que hacer será completar la información referente a la operación y, a parte, calcular los costes correspondientes.

Si por lo contrario no hay unidades disponibles en el inventario, habrá que realizar un proceso iterativo entre las tiendas en función de la distancia que haya al cliente hasta encontrar aquella que permita servir dicho pedido. Existe la posibilidad de que ninguna de las tiendas tenga el artículo buscado y que, por lo tanto, se considere la operación como una venta perdida. En este caso, se calculará el coste de la venta perdida y se actualizarán los indicadores correspondientes del sistema.

En el caso de la compra en tienda física, el procedimiento a seguir es diferente con respecto a los escenarios anteriores. De nuevo, lo primero será comprobar la disponibilidad del artículo seleccionado por el cliente. Si la disponibilidad es favorable, el procedimiento de compra será completado automáticamente añadiendo en la base de datos la información relacionada con la compra y los costes correspondientes.

Si el artículo buscado no se encuentra disponible en tienda, la novedad en este caso será la opción por parte del cliente de solicitar que le traigan una unidad del artículo deseado del inventario de una tienda o almacén diferente. Al ser una decisión no controlable por parte del empleado de la tienda, se ha modelado con una variable aleatoria que decidirá, dada esta situación, si el cliente desea o no hacer uso del servicio intertienda. Si el caso es favorable, la gestión del pedido se llevará a cabo tal y como se realiza para un pedido online. En caso contrario o que no haya disponible ninguna unidad en otra tienda, dicha operación se considera venta perdida y se contabilizará como tal.

A continuación, se muestra un resumen de los resultados obtenidos.

ORIGEN	Domicilio cliente	Tienda 1	Tienda 2	Tienda 3	Tienda 4	Tienda 5	Tienda 6	Tienda 7	Tienda 8	Tienda 9	Tienda 10	Tienda 11	Tienda 12	Tienda 13	Tienda 14	Total
Almacén	183															183
Tienda 1	16	927														943
Tienda 2	157	1	1036	5					1							1200
Tienda 3	103			973												1076
Tienda 4	270				1011					1					4	1282
Tienda 5	80					1447	1								3	1532
Tienda 6	15				1		1010									1026
Tienda 7	353							946		2						1301
Tienda 8	317								1090	3						1409
Tienda 9	89									997						1086
Tienda 10	341									2	1226					1569
Tienda 11	323	1	3	1					1	5		1476				1814
Tienda 12	858												1073			1931
Tienda 13	555													1014	12	1581
Tienda 14	626														1068	1694
Total	4286	929	1039	979	1012	1447	1011	946	1092	1010	1226	1476	1073	1014	1087	19627

Tabla 15. Distribución de pedidos origen/destino OMS. Escenario 4.

Para este último escenario, los datos presentados en la Tabla 15 son de gran utilidad debido a que se aprecia claramente el resultado de la incorporación del servicio intertienda. La Tienda 14 es aquella en la cual más se emplea con un total de 19 operaciones y, tal y como se puede ver, es la Tienda 13 aquella que surte la mayoría de dichos pedidos siguiendo el criterio de cercanía. Sin embargo, hay tiendas en las cuales no es necesario bien porque hay inventario suficiente o porque los clientes de la misma prefieran no emplear dicho servicio.

Este tipo de datos se pueden ver mejor con los siguientes indicadores.

Unidades servidas en tienda origen	15294
Unidades servidas en otra tienda (intertienda)	47
Pedidos servidos en tienda	4103
Pedidos servidos en almacén	183
Unidades servidas totales	19627
Precio medio unidad servida	94,61 €
Venta en tienda no servida (OMS)	45
Venta en tienda no servida (estimación)	1449
Pedidos online no servidos	0
Unidades totales no servidas	1494
Rotura de stock	7,59%

Como se ha comentado anteriormente, el número total de unidades servidas aumenta con respecto a los escenarios anteriores gracias a la incorporación del servicio de pedidos entre tienda. Con esto se consigue aumentar, como se ha dicho, el número de unidades servidas a la vez que se disminuye el porcentaje de rotura de stock. Por lo tanto, el objetivo fijado con la implantación de esta medida se obtiene y, con ello, la capacidad de proveer al cliente un servicio más eficiente y mejor.

Por último, el código de programación del sistema de gestión de pedidos correspondiente con este escenario se adjunta en el Anexo 7: OMS. Escenario 4.

7.8. Análisis económico

Una vez presentado los resultados fruto de las simulaciones llevadas a cabo en el modelo para cada uno de los escenarios desarrollados, el objetivo del proyecto es también analizar el impacto económico que tendrían cada una de las soluciones propuestas en un caso real. Por lo tanto, lo que se pretende es obtener los costes globales de las actividades que conforman el proceso llevado a cabo en un OMS, es decir, la gestión de los pedidos y su correspondiente distribución.

En el análisis económico que se va a realizar, el principal objetivo es el de obtener la estrategia de distribución óptimo tanto a nivel de costes como a nivel de servicio hacia el cliente. Debido

a este motivo, se tendrán en cuenta una serie de factores ajenos a los económicos que permitan evaluar también cómo de eficiente sería el modelo de cara a mejorar la experiencia del consumidor.

Para poder obtener los resultados finales del estudio se han fijado previamente una serie de costes que serán imputados dentro del modelo y que permitirán cuantificar las actividades significativas que involucra una gestión de un pedido dentro de la cadena de suministros de una compañía. Estos costes han sido obtenidos de fuentes relacionadas con el sector y, también, se han hecho algunas estimaciones para poder escalar dichos costes al tamaño de la compañía creada en este proyecto. Dichos costes se muestran en la Tabla 16 y se explican a continuación.

Coste	Valor	Unidad
Coste transporte	0,7	€/km
Coste preparación en tienda	1,78	€/unidad
Coste preparación en almacén	1,28	€/unidad
Coste de empaquetado pequeño	3	€/unidad
Coste de empaquetado mediano	5	€/unidad
Coste de empaquetado alto	7	€/unidad

Tabla 16. Costes de las actividades del OMS.

El coste de transporte, tal y como su nombre explica, es el coste que supone transportar una unidad desde cualquier punto de la red de distribución hasta su destino. Dicho coste se ha obtenido llevando a cabo una serie de estimaciones relacionadas con el consumo de un vehículo de transporte estándar, el precio de la gasolina, el número de unidades que puede transportar un lote de mercancías y, por último, el coste del transportista que trabaja para llevar a cabo la distribución de los pedidos.

El coste de preparación, tanto en tienda como en almacén, es el coste de la mano de obra de un personal encargado de llevar a cabo todo el proceso de preparación de un pedido. Este proceso implica desde la recepción hasta el momento previo del empaquetado del mismo. Este precio está desglosado para la tienda y para el almacén ya que se considera que el coste de esta actividad en la tienda es superior debido a que llevar a cabo la misma en tienda implica quitar tiempo de atención al cliente por parte del encargado que está en la tienda en ese momento.

Por último, el coste de empaquetado es el que comprende la acción de preparación de un pedido con las condiciones necesarias para poder ser enviado o transportado sin que sufra ningún tipo de desperfecto durante el momento del transporte. Dependiendo del tamaño del artículo el coste del empaquetado variará ya que se emplearán materiales y cajas de distinto tamaño y nivel de seguridad.

Los costes presentados se imputan para cada uno de los pedidos recibidos. Es cierto que no todos los pedidos tienen los mismos costes debido a que dependiendo del tipo de canal de compra se tendrán en cuenta distintos costes. Por ejemplo, para la compra física en tienda el coste de transporte siempre será cero ya que el los costes que se tienen en cuenta son únicamente del eslabón de distribución de la cadena de suministros. Todo coste ajeno a esa etapa no será tenido en cuenta, motivo por el cual costes como el de fabricación o el de transporte desde el fabricante al almacén no se tienen en cuenta de forma directa, sino que se estimarán en base a los ingresos obtenidos. Estos costes son recogidos en la tabla de pérdidas y ganancias como “Costes de Mercancías”.

Una vez presentados los costes que se han tenido en cuenta en el desarrollo de este análisis, se muestran los resultados obtenidos para cada uno de los escenarios. El formato que se le ha dado es el de una tabla de pérdidas y ganancias debido a que es la forma más común y aquella que permite visualizar y comparar todos los escenarios más cómodamente.

Por otro lado, se han desglosado los costes correspondientes a las tiendas y al almacén con el fin de poder comparar y analizar el comportamiento de ambos centros de venta en función de los criterios fijados en cada uno de los modelos.

Finalmente, mencionar que se han añadido tras los costes los resultados mostrados en los apartados anteriores referentes a cada uno de los escenarios. Lo que se pretende es analizar los costes conjuntamente con los números relacionados con los pedidos servidos, venta perdida, etc.

	E1	E2	E3	E4
Ingresos en tienda	1.477.581,23 €	1.446.548,54 €	1.470.058,17 €	1.473.902,57 €
Ingresos online	344.532,59 €	383.036,54 €	383.036,84 €	383.036,84 €
Ingresos totales	1.822.113,82 €	1.829.585,08 €	1.853.095,01 €	1.856.939,41 €
Coste Mano de Obra Tienda	27.387,00 €	34.718,00 €	34.874,00 €	34.945,00 €
Coste Mano de Obra Almacén	5.486,00 €	213,00 €	234,00 €	234,00 €
Coste Packaging Tienda	- €	16.351,00 €	16.351,00 €	16.714,00 €
Coste Packaging Almacén	15.541,00 €	603,00 €	603,00 €	603,00 €
Coste Transporte Tienda	- €	390.646,00 €	478.768,00 €	484.937,00 €
Coste Transporte Almacén	993.269,00 €	15.698,00 €	21.353,78 €	21.353,78 €
Coste Rotura Stock Tienda	136.785,60 €	170.121,99 €	145.021,80 €	141.347,34 €
Coste Rotura Stock Almacén	34.833,60 €	- €	- €	- €
Coste de Venta	1.213.302,20 €	628.350,99 €	697.205,80 €	699.953,34 €
Coste de Mercancías	728.845,53 €	731.834,03 €	741.238,00 €	742.775,76 €
Coste Total	1.942.147,73 €	1.360.185,02 €	1.438.443,80 €	1.442.729,10 €
Margen Bruto	-120.033,91 €	469.400,06 €	414.651,21 €	414.210,31 €

	E1	E2	E3	E4
Unidades servidas en tienda origen	15386	15038	15302	15294
Unidades servidas en otra tienda (intertienda)	0	0	0	47
Pedidos servidos en tienda	0	4119	4103	4103
Pedidos servidos en almacén	3917	167	183	183
Unidades servidas totales	19303	19324	19588	19627
Precio medio unidad servida	94,40 €	94,67 €	94,60 €	94,61 €
Venta en tienda no servida (OMS)	0	348	84	45
Venta en tienda no servida (estimación)	1449	1449	1449	1449
Pedidos online no servidos	369	0	0	0
Unidades totales no servidas	1818	1797	1533	1494
Rotura de stock	9,24%	9,13%	7,79%	7,59%

Escenario 1

Partiendo de los resultados presentados para el Escenario 1 en los apartados anteriores, es de esperar que sus ingresos estén por debajo de la media de los demás escenarios ya que es aquel que presenta mayor porcentaje de rotura de stock y menor número total de pedidos servidos. Esto se debe fundamentalmente al número de pedidos procedentes de la tienda online que no se han podido abastecer a causa de la estrategia tomada en dicho escenario. Depender únicamente del inventario de un centro de almacenamiento puede dar lugar a un servicio ineficiente debido a la exactitud requerida en el cálculo del inventario que debe cubrir la demanda.

Como consecuencia también de la estrategia tomada, los costes relacionados con la actividad en el almacén son bastante elevados, sobre todo el de transporte. Los pedidos a domicilio son todos servidos por el inventario del almacén y, por este motivo, la cifra es tan elevada. Es cierto que el almacén está situado geográficamente con el objetivo de optimizar dicho coste. Por lo tanto, uno de los aspectos que se debería de tener en cuenta a la hora de llevar a cabo un criterio de distribución como el tomado para este escenario sería el de analizar previamente la red logística con el fin de ver si es eficiente o no optar por este modelo de distribución ya que implica unos altos costes de transporte que podrían penalizar bastante los resultados de la compañía.

El resto de costes para este modelo en comparación con el de transporte en el caso del almacén no son prácticamente significativos. El correspondiente con la rotura de stock sería el siguiente que tiene mayor peso. Dicho coste se ha calculado de la misma forma para todos los escenarios, es decir, se ha tomado la estimación realizada de venta perdida en tienda y junto con el precio

media de unidad servida se ha realizado dicho cálculo. Hay que tener en cuenta que es una estimación, por lo cual puede haber cierta desviación en dicho valor en caso de implantar esta solución que se está proponiendo.

Los costes de mercancías en el modelo desarrollado se han calculado también mediante una estimación. Al no tener en la simulación datos relacionados con los procesos previos a la distribución en la cadena de suministro, se ha optado por establecer un porcentaje que representa la parte correspondiente de los ingresos totales que se corresponde con los procesos de fabricación, transporte interno, etc. El porcentaje establecido es un 40% de los ingresos totales obtenidos en el período simulado.

Teniendo en cuenta los costes de venta y los costes de mercancías, el resultado final obtenido para este escenario resulta ser negativo, es decir, el margen bruto toma un valor por debajo de cero. Por este motivo, cabe pensar que el criterio de diseño para la gestión de los pedidos en este escenario no es eficiente desde el punto de vista económico. Como se mencionaba en el Apartado 2.3: Futuras tendencias del sector, las compañías del sector retail están tratando de centralizar el inventario de todos sus canales de venta con el objetivo de mejorar la eficiencia y, sobre todo, los costes de operación. Debido a esto surge el concepto de omnicanalidad que en la actualidad se encuentra tan presente en los modelos de gestión de las empresas.

Este tipo de modelo de gestión de los pedidos y del inventario podría ser rentable en redes de distribución mayores en las que el número de almacenes fuese mayor. De esta forma, se podría disminuir el radio de alcance de cada uno de los almacenes y, con ello, el coste de transporte que es el que penaliza verdaderamente el coste total de operación.

Escenario 2

En este nuevo escenario los ingresos totales han aumentado con respecto a los del escenario anterior. El motivo fundamental de dicho crecimiento es que existe mayor flexibilidad en el sistema para poder cubrir la demanda a causa de la configuración del modelo. Ahora los pedidos de la tienda online se pueden abastecer con el inventario de cualquiera de las tiendas siguiendo los criterios de cercanía y de disponibilidad de inventario. Por tanto, al ser mayor el número de pedidos servidos en este escenario, los ingresos totales serán mayores. Cabe destacar que los ingresos de la tienda online han aumentado mientras que los procedentes de la tienda física han disminuido al haber aumentado la venta perdida en dicho canal de venta.

Los costes de venta en este caso se reducen considerablemente debido a que, como consecuencia de la implementación del criterio de cercanías, el coste de transporte disminuye bastante respecto al caso anterior. Se ha pasado de un modelo de gestión de pedidos muy estático, en el cual todo el inventario se encontraba centralizado en un único centro, a un

modelo más dinámico que permite la elección más apropiada de las unidades que se le asignarán a cada una de las líneas de pedidos para ser servidas.

Por otro lado, el reparto de los costes en este escenario varia ya que la actividad en el almacén se reduce bastante debido a que se lleva a cabo la redistribución del inventario del almacén a lo largo del resto de centros de almacenamiento de la red de distribución. El número de pedidos servidos por el almacén se ve reducido considerablemente y, por lo tanto, disminuyen todos los costes relacionados con la actividad en el mismo. Como consecuencia, el coste de operación en las tiendas pasa a ser ahora muy superior con respecto al escenario anterior.

Cabe destacar la fuerte disminución de costes a nivel global que se ha producido respecto al caso anterior, sobre todo el coste de transporte que es el que tiene mayor peso en este modelo. Este hecho da como resultado un sistema de gestión de pedidos mejor a nivel económico, aunque a nivel de eficiencia presenta unos resultados que no varían significativamente respecto a lo presentado en el Escenario 1. Los niveles de venta perdida se revierten penalizando ahora la venta en tienda física. El número de unidades no servidas es casi el mismo y, por lo tanto, el porcentaje de rotura de stock también lo es.

Como resultado final de este escenario, los costes de venta de experimentado una disminución considerable con respecto al anterior tal y como se ha comentado anteriormente. Por otro lado, los costes de mercancías aumentan al haber aumentado los ingresos. Como se mencionó para el caso anterior, el porcentaje establecido para definir estos costes se mantiene. Finalmente, que los costes hayan decrecido provoca que el margen bruto sea bastante favorable y que, a nivel económico, este escenario sea muy positivo.

Escenario 3

Como se comentó en el Apartado 7.7.4: Escenario 3, la principal variante de este nuevo escenario con respecto al anterior es la incorporación del inventario de seguridad en cada una de las tiendas de la red de distribución. El motivo fundamental de este nuevo criterio de diseño del OMS es la implantación de una medida contra la penalización provocada sobre el número de unidades no servidas en tienda física. La situación en el segundo escenario prácticamente se revierte, tal y como se mencionó anteriormente, favoreciendo la venta online a la vez que se penaliza el otro medio de venta.

Por lo tanto, como consecuencia de la medida tomada, el número total de unidades servidas aumenta con respecto a los casos anteriores y, por este motivo, los ingresos totales también lo hacen. La diferencia principal es que el número de unidades vendidas en tienda ha aumentado gracias al establecimiento del stock de seguridad, permitiendo una mayor disponibilidad de unidades en tienda dedicadas únicamente a la venta física. Ciertamente, la medida tomada sirve

para evitar que el efecto de la temporalidad con la que los pedidos online se solicitan se pueda controlar en cierto grado ya que es un hecho que no se puede predecir plenamente.

Otra variación con respecto a los casos anteriores sería el aumento de costes que implica la introducción del stock de seguridad en las tiendas y lo que esto conlleva. Los costes de mano de obra y de empaquetado, tanto en tienda como en la plataforma de venta online, no sufren una variación considerable. Sin embargo, el coste de transporte aumenta en ambos canales de venta ya que el número de unidades servidas en total aumenta.

Cabe destacar que la actividad en el almacén a la hora de cubrir pedidos online es más alta debido a que el stock de seguridad en las tiendas hace que no haya disponibilidad para cubrir desde las tiendas más cercanas el pedido solicitado. Este hecho provoca un aumento de coste de transporte considerable debido a que la distancia de los domicilios de los clientes al almacén probablemente es mayor que desde las tiendas.

Por otro lado, a causa del aumento de los ingresos por el incremento del número de unidades servidas, el coste de mercancías aumenta, al ser un porcentaje del mismo. Como este coste aumenta y el coste de operación también lo hace, los costes serán más altos a nivel global. Como consecuencia, se puede comprobar que el aumento de los ingresos obtenido en este escenario no es suficiente como para cubrir el aumento de costes que implica la implantación de la solución propuesta.

El resultado final del escenario presentado es un modelo de gestión de pedidos más eficiente a nivel de operación, reflejado en la disminución considerable del porcentaje de rotura de stock, pero con un margen bruto menor que el del escenario anterior, el cuál no tenía incorporado el stock de seguridad.

Escenario 4

El último escenario llevado a cabo en la simulación del sistemas de gestión de pedidos o, dicho de otra forma, el OMS incorpora el servicio intertienda. Con esta implementación lo que se pretendía era aumentar el número de unidades vendidas en tienda aprovechando el stock disponible en otras tiendas de la red de distribución. Este hecho se ve directamente reflejado en los ingresos totales mostrados en la cuenta de pérdidas y ganancias de este escenario. Un total de 39 unidades adicionales que se consiguen vender provoca aproximadamente un 0,2% de aumento en los ingresos en tienda física.

A nivel de costes, el coste de mano de obra aumenta, como se esperaba, debido al mayor nivel de actividad en las mismas y la necesidad de un mayor tiempo de trabajador para cubrir la demanda. Este coste es directamente proporcional al número de unidades servidas en dicho canal de venta. Lo mismo ocurre con el coste de empaquetado, aunque para el criterio de diseño añadido en este escenario, dependerá también del número de unidades servidas a través del

servicio de intertienda. Las unidades que sean pedidas serán preparadas y empaquetadas, con lo cuál este coste también se verá incrementado.

El servicio intertienda requiere del uso de transporte interno de la empresa para poder trasladar las unidades solicitadas de una tienda a otra. Por esta razón, la implantación de esta medida también implica un coste adicional, aunque no deberá ser muy elevado ya que se emplea también el criterio de cercanía para poder abastecer este tipo de demanda. En la cuenta de pérdidas y ganancias se refleja este incremento, que se corresponde con un 1,2% de aumento del coste con respecto al escenario anterior.

El principal efecto de la implantación de la medida presentada para el escenario en cuestión es la reducción considerable del porcentaje de rotura de stock, que contrarresta el aumento del resto de costes debido a la prestación del servicio. Existe una disminución en torno al 0,2% en los niveles de venta perdida. Una posible conclusión de la simulación de este escenario es que la implantación de un servicio que mejora la eficiencia del sistema de operación de la compañía implica un aumento de costes internos que son inevitables y que deberán de reducirse lo máximo posible si se quiere que la idea sea rentable.

Como resumen de los costes, el coste de operación o, también llamado, coste de venta aumenta por los motivos expuestos anteriormente, al igual que también lo hacen los costes de mercancías a causa del aumento de los ingresos totales de la compañía. Al igual que en los escenarios anteriores, los costes de mercancías representan un porcentaje definido, e igual en todos los escenarios, sobre los niveles de ingresos producidos por la venta de los artículos que conforman el catálogo de la compañía.

Por lo tanto, el margen bruto en este escenario queda prácticamente igual que en el escenario anterior, siendo ligeramente inferior ya que el aumento de los ingresos totales no llega a cubrir por completo el incremento de costes obtenido.

Análisis general

Dentro de los cuatro escenarios presentados en la simulación llevada a cabo para el sistema de gestión de pedidos, se puede diferenciar claramente entre dos tipos de estrategias de configuración de una red de distribución de una compañía del sector.

En primer lugar, un sistema centralizado en el que el inventario empleado para cubrir la demanda correspondiente de la plataforma de venta online se encuentra en un almacén de la red logística. Con esto lo que se pretende es diferenciar ambos canales de venta y que cada uno pueda gestionar de manera independiente sin necesidad de controlar ambos inventarios a la vez.

Esta metodología implica un alto coste de transporte que hace que los resultados obtenidos no sean positivos. El hecho de servir los pedidos a domicilio desde un punto concreto de la red

hace que las distancias sean muy grandes y que el coste se vea fuertemente perjudicado. Por este motivo, este criterio de diseño podría ser útil en el caso de tener una red de distribución más amplia en la que el número de almacenes fuese superior, pudiendo cubrir más eficientemente los puntos de demanda.

En segundo lugar, el sistema presentado presenta una configuración del inventario descentralizada en la que las tiendas también pueden servir los pedidos de la plataforma *e-commerce*. Dentro de este sistema se añaden una serie de complementos o medidas que permiten mejorar el comportamiento del modelo, siendo estas el stock de seguridad y el servicio intertienda.

El escenario básico de esta configuración presenta el mejor resultado a nivel económico ya que los costes son inferiores al haber un menor número de unidades vendidas y, por lo tanto, un menor nivel de actividad en la red de distribución. Sin embargo, es un sistema menos eficiente ya que hay un mayor porcentaje de rotura de stock que penaliza la venta en tienda física.

Como consecuencia de los resultados obtenidos, tal y como se comentó en la explicación del Escenario 3, es necesario implementar una serie de medidas que permiten mejorar los valores resultantes para las variables relacionadas con la mejora de la experiencia del cliente en el proceso de compra. Como es lógico, la implantación de estos nuevos factores implica un aumento de los costes de producción y de operación debido al uso de un mayor número de recursos. Este aumento de costes tiene repercusión sobre el resultado final económico de los modelos en los que se implementan dichas medidas.

En resumen, la implementación o mejora de nuevas técnicas que proveen un mejor rendimiento en la gestión de los procesos y, sobre todo, una mejor experiencia para los clientes puede acarrear una inversión fuerte inicial. Sin embargo, el servicio prestado, generalmente, tenderá a generar un aumento de la demanda gracias a la satisfacción del consumidor y todo lo que ello conlleva. Por este motivo, el Escenario 4 sería el óptimo en el estudio llevado a cabo en este modelo de optimización ya que pese a que se sacrifica en cierta medida los resultados a nivel económico, la eficiencia y la imagen de cara al cliente será muy favorable, lo cual provocará una mejora de los ingresos en vistas a un futuro a corto plazo.

8. Desarrollo del modelo de predicción

8.1. Introducción

El segundo modelo que conforma este proyecto está basado en técnicas de Inteligencia Artificial haciendo uso de herramientas de Machine Learning, con las cuáles se pretende realizar una predicción de la demanda teniendo en cuenta datos históricos generados específicamente para ello.

Partiendo de la estructura de la red de distribución definida en el modelo de optimización explicado en el apartado anterior, se han llevado a cabo una serie de modificaciones sobre las bases de datos iniciales con el fin de sesgar los datos de entrada a un modelo muy específico de acuerdo con las variables que se quieren medir y que son necesarias para realizar la predicción.

La generación de datos se explicará con detalle en el siguiente apartado, pero la idea general es generar los pedidos correspondientes a tres temporadas de invierno para tres tipos de artículos, dos de ellos propios de la temporada, como son un abrigo y un chubasquero, y el otro sería una camisa, prenda estándar a nivel de moda. Estos tres artículos conforman nueve referencias SKU para el total de las tres temporadas, es decir, existen tres modelos diferentes de cada uno de los artículos que son propios de cada una de las temporadas.

El objetivo del desarrollo de este modelo es el de analizar las ventas a través del histórico de cada una de las SKU con el fin de predecir las ventas para un futuro próximo y, además, poder determinar una estrategia que permita estudiar las preferencias del cliente en cuanto al diseño de nuevos modelos para cada uno de los tres artículos definidos.

Por otro lado, la generación de datos se ha llevado a cabo teniendo en cuenta dos factores primordiales en el sector, el clima y las promociones. Más adelante se comentará con detalles la implementación de estos dos factores en el modelo pero, básicamente, la intención es analizar cómo afectan ambos en las ventas de cada tipología de artículo y el impacto que tienen sobre el nivel de ingresos en general.

Para finalizar, se implementarán dos modelos de predicción distintos, en primer lugar un modelo ARIMA (del inglés *Autoregressive Integrated Moving Average*) y el otro sería un modelo de predicción de series temporales desarrollado por el equipo de Facebook's Core Data Science, denominado Prophet. Este último modelo está basado en un modelo aditivo aplicable a tendencias no lineales. Teniendo en cuenta estos dos modelos de forma independiente, se analizarán los resultados de ambos sobre los datos reales que se van a generar y se desarrollará en alto detalle aquel que presente mayor precisión con el objetivo de determinar una predicción a futuro para las próximas temporadas teniendo en cuenta los factores mencionados anteriormente.

8.2. Generación de datos

Tal y como se ha comentado en el apartado de introducción, la generación de los pedidos sigue un procedimiento muy similar al empleado en el modelo de optimización desarrollado también en este proyecto. Para ello, se definirán una serie de datos de entrada, como son los artículos, las tiendas, los clientes, etc. y se desarrollará posteriormente un código de generación aleatoria en el lenguaje de programación Python, que permitirá obtener una base de datos de salida con los pedidos diarios para el período de tiempo deseado según una serie de parámetros que determinan y caracterizan a la misma vez el día a día del comercio diseñado objeto de este proyecto.

En primer lugar, se redujo el número de tiendas y clientes en comparación con el modelo anterior de cara a trabajar con una red de distribución más simple y asequible en relación con el número de parámetros a emplear. Por lo tanto, en cuanto a las tiendas, se tomó la decisión de acotar la red de distribución a la ciudad de Madrid en la que ya había definidas cuatro tiendas, empleadas en el anterior modelo.

ID Tienda	Tienda	Ciudad	Provincia	Localización	Código Postal	Latitud	Longitud
1	Tienda 1	Madrid	Madrid	Plaza del Carmen, 6	28013	40,4185927	-3,7034448
2	Tienda 2	Madrid	Madrid	Calle Azafrán, 15	28023	40,4500924	-3,7661171
3	Tienda 3	Madrid	Madrid	Gran Vía, 102	28013	40,4232249	-3,7105843
4	Tienda 4	Madrid	Madrid	Calle Serrano, 85	28006	40,4366852	-3,6865791

Tabla 17. Tiendas de la red de distribución. Modelo de predicción.

El siguiente paso en la definición de los parámetros de entrada del modelo fue la creación de la base de datos de los clientes. Para este caso, no fueron necesarias características de género y localización de cada uno de los clientes ya que la idea en este modelo es que los clientes fuesen tiendas minoristas que van a vender los productos desarrollados por el negocio creado, lo cuál facilita la lógica de programación del modelo.

Id Cliente	Cliente	Id Cliente	Cliente
1	Cliente 1	6	Cliente 6
2	Cliente 2	7	Cliente 7
3	Cliente 3	8	Cliente 8
4	Cliente 4	9	Cliente 9
5	Cliente 5	10	Cliente 10

Tabla 18. Clientes. Modelo de predicción.

Tras haber definido las tiendas y los clientes, lo siguiente que es necesario definir son los artículos que van a conformar el catálogo de cada una de las temporadas de invierno que se van a simular. Como se comentó en la introducción, el objetivo es estudiar el histórico de cada uno

de los modelos de los artículos en sus temporadas correspondientes y ver que influencia tiene sobre las ventas las características que se incorporan en cada una de las prendas año tras año. Estos artículos se muestran en la Tabla 19, en la cuál se presentan las características de cada uno de ellos.

SKU	Id Artículo	Artículo	Temporada	Dimensión Paquete	Color	Precio	Coste
1001	1	Chubasquero Azul Largo con Capucha	1	M	Azul	149,90 €	52,47 €
1002	2	Chubasquero Beige con Botones	2	M	Beige	159,95 €	55,98 €
1003	3	Chubasquero Negro Corto con Capucha	3	M	Negro	125,00 €	43,75 €
1004	4	Camisa Slim Lisa Azul	1	P	Azul	79,00 €	27,65 €
1005	5	Camisa Regular de Rayas Azul	2	P	Azul	65,00 €	22,75 €
1006	6	Camisa Slim Lisa de Cuadros Verde	3	P	Verde	69,95 €	24,48 €
1007	7	Chaquetón Gris con Doble Botón	1	G	Gris	199,95 €	69,98 €
1008	8	Chaquetón Azul con Capucha	2	G	Azul	179,99 €	63,00 €
1009	9	Chaquetón de Pana Marrón	3	G	Marrón	199,95 €	69,98 €

Tabla 19. Artículos de las tres temporadas de invierno. Modelo de predicción.

Como se puede apreciar en la tabla, existen tres grupos diferentes de artículos dentro de los cuales hay diferentes modelos, formando un total de nueve referencias distintas en el catálogo general. Dos de los grupos son prendas características de la temporada de invierno, como son los chubasqueros y los abrigos. La venta de estos artículos se ve influenciada por los parámetros meteorológicos, los cuales se pretenden implementar en el modelo con el fin de ver su efecto sobre la predicción de las ventas. Por otro lado, las camisas son una prenda más estándar en el mercado, cuyo nivel de ventas no se ve influenciado por el clima pero si que lo está por otro factor que se pretende incorporar, como son las promociones y las festividades propias de la temporada de invierno, por ejemplo el Black Friday o las compras de Navidad.

Por otro lado, se han incorporado en esta tabla el coste de producción de cada una de las referencias con el fin de analizar en cada una de las temporadas no sólo los ingresos y el número de ventas, sino también el margen sobre beneficios de cada una de ellas a lo largo de los tres períodos simulados.

Tras haber definido las tres tablas de datos de entrada al modelo que caracterizan el negocio diseñado, el siguiente paso sería analizar los factores que se pretenden tener en cuenta en el modelo, como son el clima y las promociones y festividades. El objetivo de este análisis es poder definir posteriormente el nivel de venta de cada una de las tiendas y la distribución diaria de ventas de cada una de las referencias en función de los parámetros considerados. Para ello, se hará uso de dos fuentes de datos donde se recopilan tanto el histórico meteorológico diario para cada uno de los períodos simulados, como el histórico de búsquedas en Internet de cada una de las prendas. Estas fuentes de datos son Aemet (Agencia Estatal de Meteorología) y Google Trends respectivamente.

En primer lugar, desde Aemet se obtuvieron los datos referentes a la temperatura y el nivel de precipitaciones para cada uno de los días que se pretenden caracterizar en el modelo, mientras que de Google Trends los datos que se obtienen son los niveles de búsqueda de cada uno de los artículos definidos, dato que tiene una fuerte relación con las ventas de dichas prendas según varios análisis realizados por profesionales del sector.

A continuación, se muestran los datos climáticos para cada una de las temporadas en base a la información obtenida por la Agencia Estatal de Meteorología. Dicha información se comparará posteriormente con la obtenida por Google Trends de cara a analizar si existe correlación entre ambas fuentes de datos en relación con las ventas y la influencia de los parámetros que se quieren introducir en el modelo.

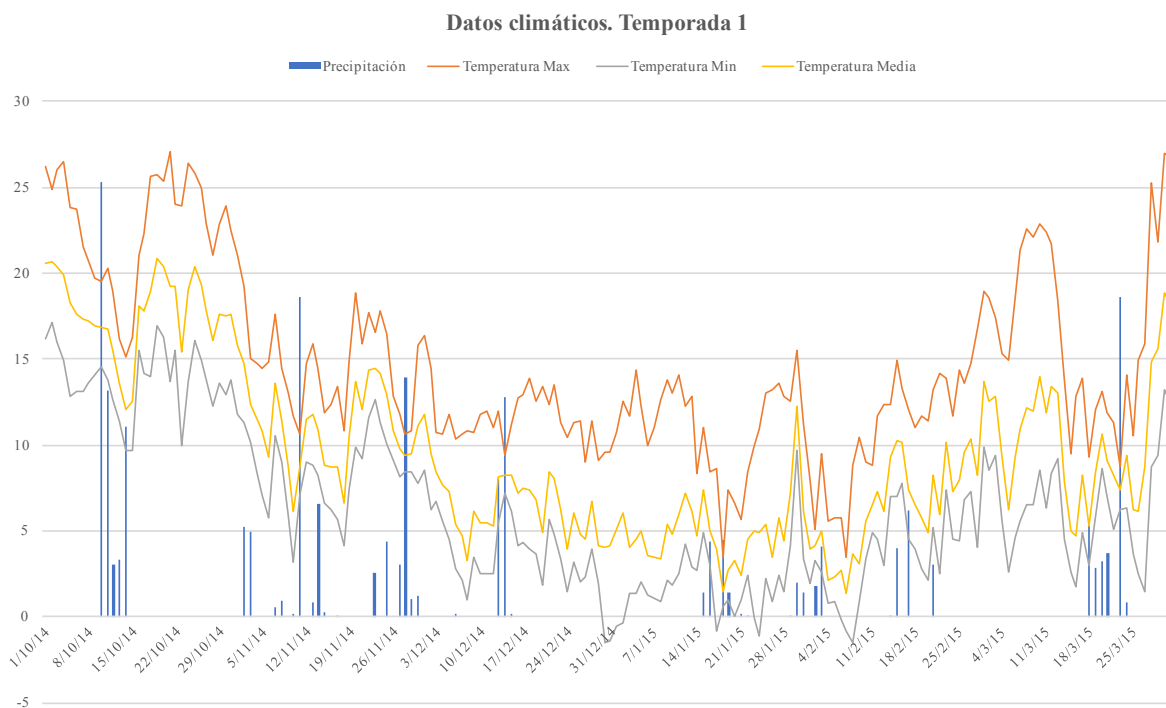


Figura 30. Datos climáticos de la temporada 1.

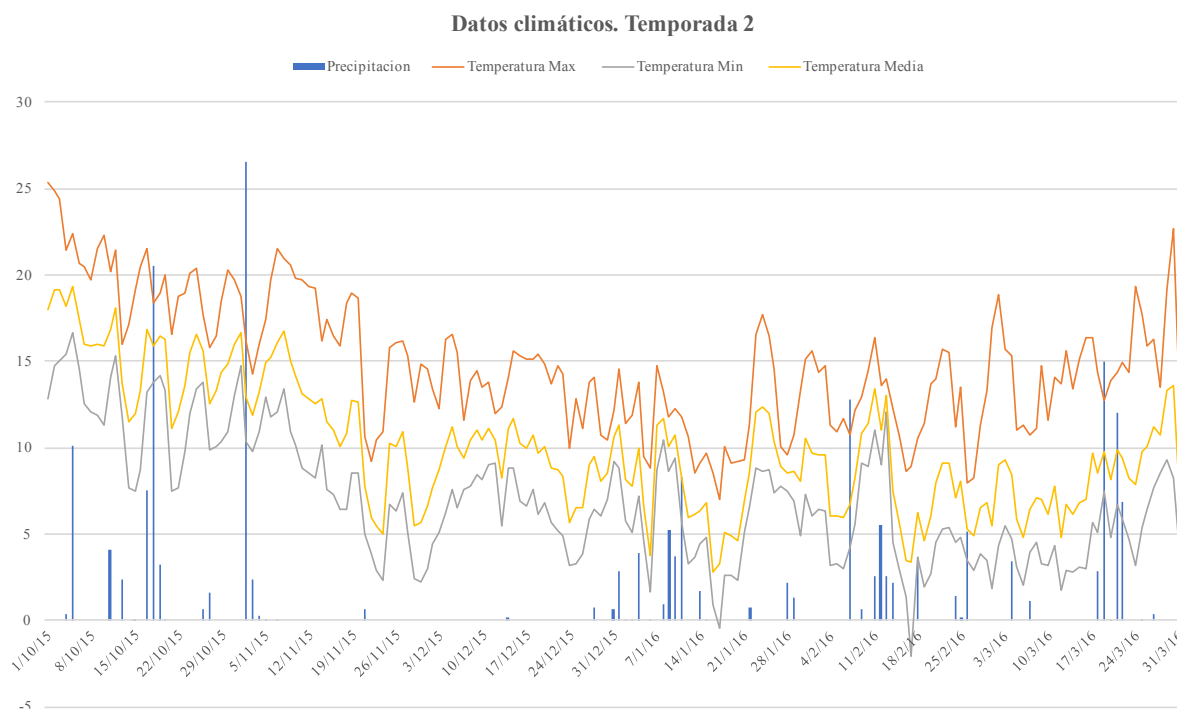


Figura 31. Datos climáticos de la temporada 2.

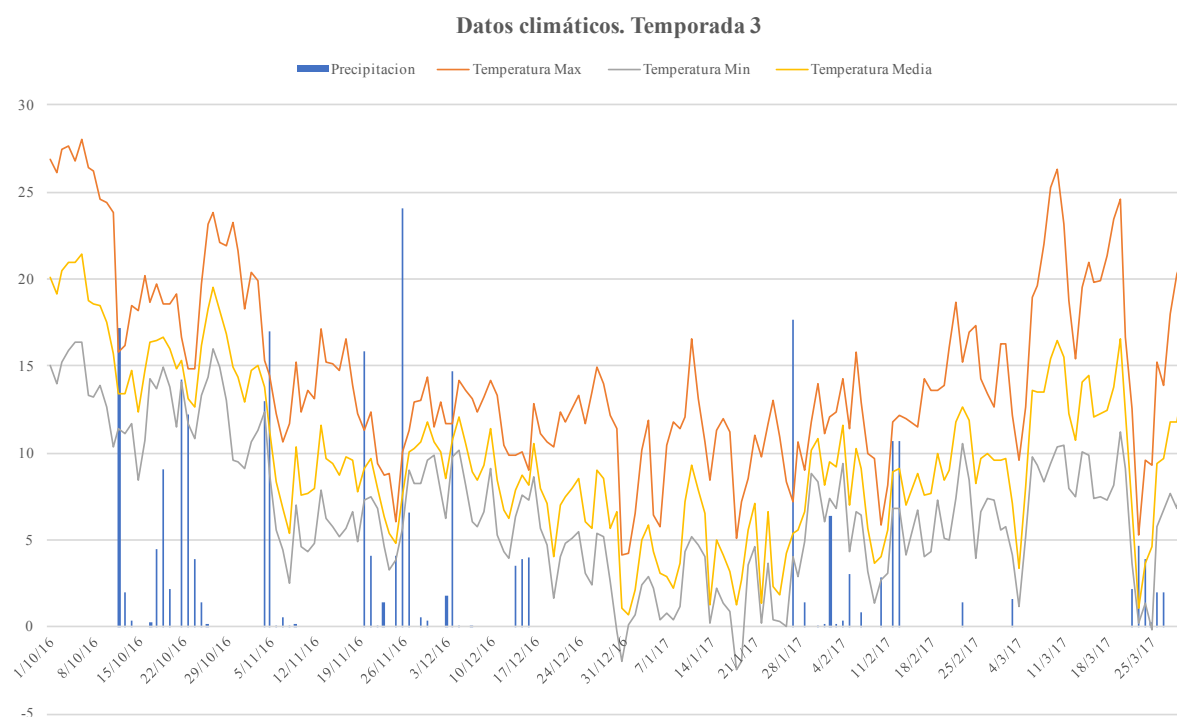


Figura 32. Datos climáticos de la temporada 3.

Una vez definidas las variables meteorológicas en base a los parámetros obtenidos de las bases de datos de Aemet, lo siguiente que habría que definir serían las promociones. Para ello, se presentan a continuación los calendarios de cada una de las temporadas con sus correspondientes eventos y sus porcentajes de rebajas establecidos.



Figura 33. Calendario de eventos. Temporada 1.

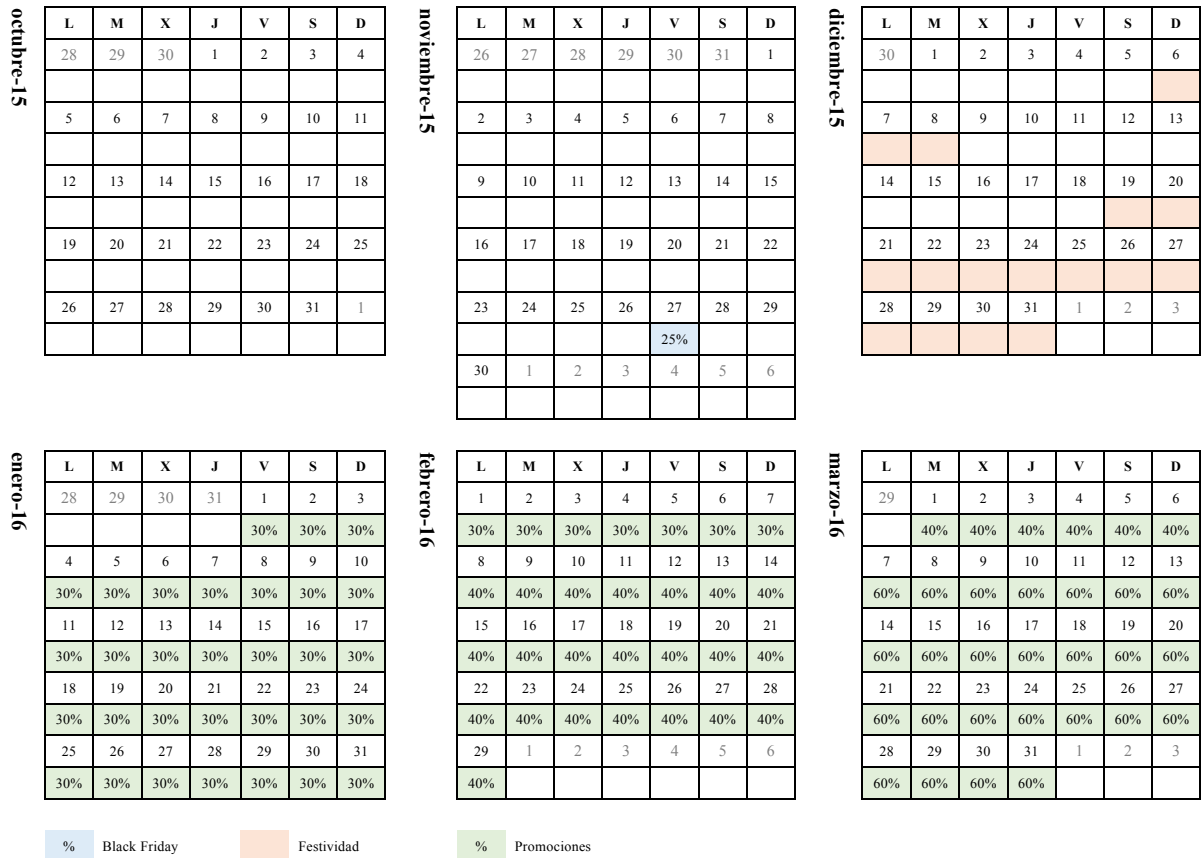


Figura 34. Calendario de eventos. Temporada 2.



Figura 35. Calendario de eventos. Temporada 3.

Considerando la información mostrada en las figuras anteriores, se puede apreciar como los eventos principales remarcados son los siguientes:

- Black Friday.
- Festividad de los días de la Inmaculada y la Constitución.
- Vacaciones de Navidad.
- Rebajas de la temporada de invierno. En dichas rebajas se ha tenido en cuenta la variación del porcentaje de rebajas a lo largo de todo el período, finalizando con un valor elevado característico de las semanas en las que el objetivo es agotar el inventario.

Finalmente, los últimos valores relacionados con el histórico son los mencionados anteriormente y obtenidos de la base de datos de Google Trends. Con el objetivo de observar la relación que existe entre el número de búsquedas y los dos factores de nuestro modelo (clima y promociones), se presentan los datos desglosados en varias gráficas caracterizadas para cada tipo de artículo y para cada una de las temporadas. En dichas gráficas se muestran los datos climáticos y el número de búsquedas en la plataforma.



Figura 36. Estudio de la relación entre el comportamiento del cliente y el clima.

Como se puede ver analizando las gráficas, existe una relación entre el comportamiento del consumidor y el clima ya que, por ejemplo, en el caso de los chubasqueros los clientes tienden a comprar dicha prenda en el momento en el que empieza la temporada de lluvia. Este comportamiento se repite para cada una de las tres temporadas. Por otro lado, En el caso de los abrigos, los niveles de venta aumentan en el momento en el que las temperaturas disminuyen y, además, se puede apreciar como el número de ventas aumenta significativamente cuando llega el período de promociones, al igual que ocurre con las camisas. Por lo tanto, se puede decir que existe correlación entre los dos parámetros considerados en el modelo y el comportamiento del consumidor de cara a las ventas.

Teniendo en cuenta que la hipótesis realizada es correcta, el siguiente paso en la generación de los datos del modelo es establecer los niveles de venta diarios para cada una de las tiendas y tipología de artículos. Para ello, el primer paso es definir los ingresos mensuales en cada una de las tiendas para cada uno de los meses que se pretenden simular. En la Tabla 20, se muestran los valores establecidos.

Período	Fecha	Tienda 1	Tienda 2	Tienda 3	Tienda 4
1	10/14	8.680 €	8.567 €	9.745 €	12.280 €
2	11/14	13.888 €	14.136 €	18.516 €	22.718 €
3	12/14	15.972 €	16.257 €	21.294 €	26.125 €
4	1/15	17.569 €	17.882 €	23.423 €	28.738 €
5	2/15	10.541 €	10.729 €	14.054 €	17.243 €
6	3/15	8.433 €	8.583 €	11.243 €	13.794 €
7	10/15	7.812 €	7.711 €	8.771 €	11.052 €
8	11/15	12.499 €	12.723 €	16.665 €	20.446 €
9	12/15	14.374 €	14.631 €	19.164 €	23.513 €
10	1/16	15.812 €	16.094 €	21.081 €	25.864 €
11	2/16	9.487 €	9.656 €	12.649 €	15.518 €
12	3/16	7.590 €	7.725 €	10.119 €	12.415 €
13	10/16	9.375 €	9.253 €	10.525 €	13.262 €
14	11/16	14.999 €	15.267 €	19.998 €	24.535 €
15	12/16	17.249 €	17.557 €	22.997 €	28.215 €
16	1/17	18.974 €	19.313 €	25.297 €	31.037 €
17	2/17	11.385 €	11.588 €	15.178 €	18.622 €
18	3/17	9.108 €	9.270 €	12.143 €	14.898 €

Tabla 20. Ventas mensuales por tienda. Modelo de predicción.

Una vez definidos los ingresos de cada una de las tiendas, la idea es establecer una parametrización diaria de cada tipología de artículo con ayuda de los datos obtenidos de Google Trends con el fin de caracterizar el día a día del negocio y establecer la relación que dichos datos tienen con el clima y las promociones sobre las ventas definidas. Tras todo este

proceso de parametrización, el resultado obtenido es una tabla en la cual se expresan las ventas diarias en cada una de las tiendas.

Fecha	Tienda 1	Tienda 2	Tienda 3	Tienda 4	Total
1/10/14	332 €	327 €	372 €	469 €	1.501 €
2/10/14	328 €	324 €	369 €	465 €	1.486 €
3/10/14	322 €	318 €	361 €	455 €	1.456 €
4/10/14	328 €	324 €	369 €	465 €	1.486 €
5/10/14	345 €	340 €	387 €	488 €	1.560 €
6/10/14	289 €	285 €	325 €	409 €	1.308 €
7/10/14	224 €	221 €	251 €	317 €	1.013 €
8/10/14	209 €	206 €	235 €	296 €	945 €
9/10/14	319 €	314 €	358 €	451 €	1.441 €
10/10/14	224 €	221 €	251 €	317 €	1.013 €
...	...	...	...	...	...
26/3/17	331 €	337 €	441 €	541 €	1.650 €
27/3/17	283 €	288 €	377 €	463 €	1.411 €
28/3/17	261 €	266 €	348 €	427 €	1.302 €
29/3/17	275 €	279 €	366 €	449 €	1.369 €
30/3/17	333 €	339 €	444 €	544 €	1.660 €
31/3/17	251 €	256 €	335 €	411 €	1.254 €

Tabla 21. Ventas diarias por tienda. Modelo de predicción.

Gracias a la parametrización realizada, en el momento en el que se ejecute el código de programación que da lugar a la generación de cada uno de los pedidos, estos niveles de ingresos darán lugar a un número de unidades de cada una de las referencias del catálogo para cada una de las tiendas de manera diaria. Por lo tanto, los factores de clima y promociones estarán ya incluidos de forma interna en la generación de los datos y, como consecuencia, en el modelo.

Una vez que están todos los datos de entrada definidos, la generación de los pedidos se lleva a cabo de forma directa haciendo uso de la lógica de programación diseñada en la que se realiza una creación aleatoria de una base de datos de pedidos de una forma muy similar a la empleada en el modelo de optimización. Dicha lógica se presenta en el esquema de principios mostrado en la Figura 37. A continuación, se presenta un extracto de la tabla de pedidos mencionada y una serie de gráficas donde se muestran los resultados obtenidos.

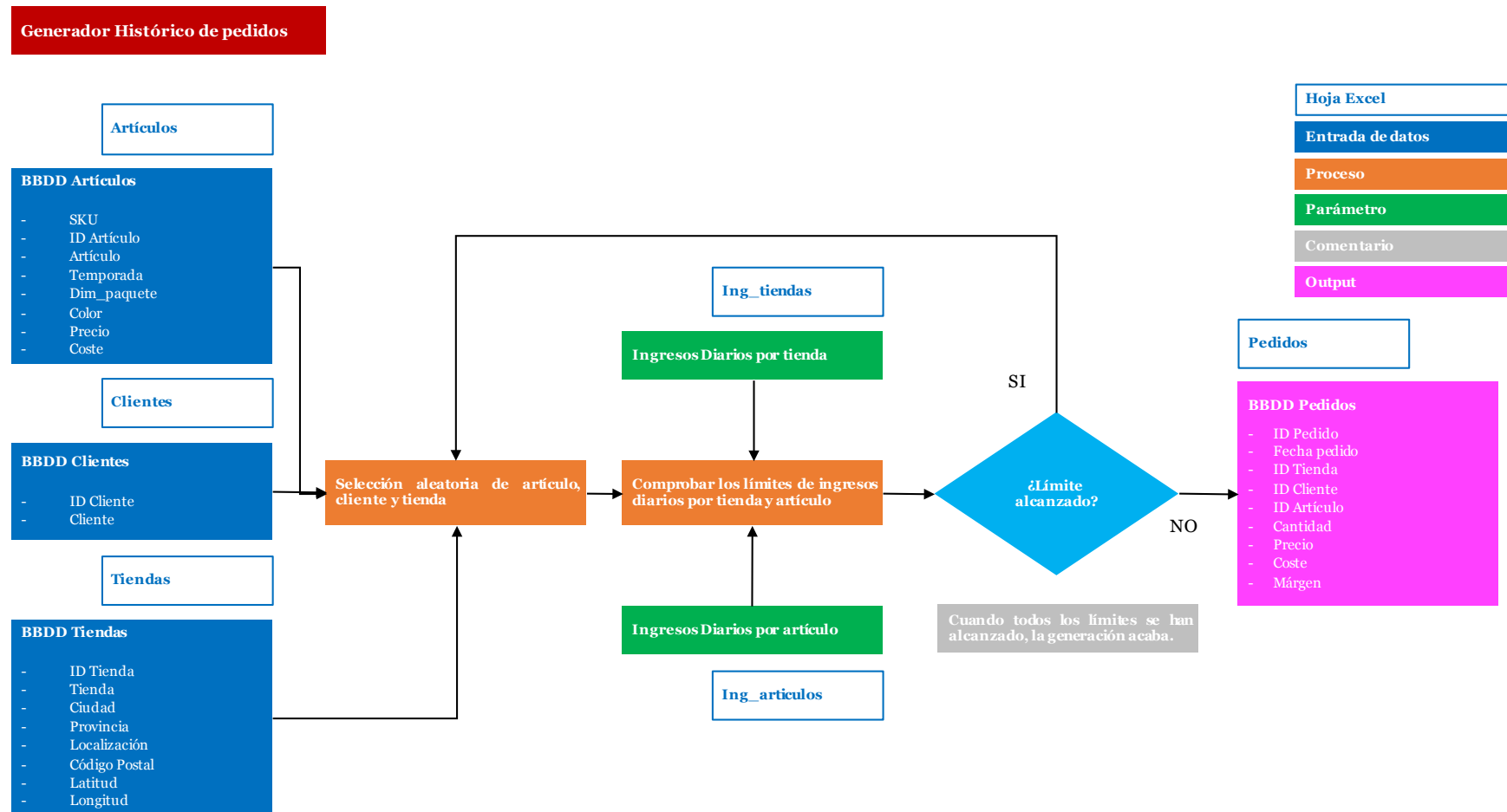


Figura 37. Esquema de principios de la lógica del generador del histórico de pedidos.

Fruto de la resolución de la lógica de programación del generador de pedidos presentado hasta ahora, se obtiene una tabla de salida en la cuál se encuentran los pedidos para cada una de las tiendas con una frecuencia diaria en base a los criterios definidos y los factores que caracterizan al modelo en cuestión. A continuación, se muestra un extracto de la tabla mencionada, de la cuál se ha llevado a cabo un informe con la ayuda del programa Power BI de Microsoft de donde se han extraído las gráficas mostradas en la Figura 38 y Figura 39 respectivamente.

Id Pedido	Fecha	Id Tienda	Id Cliente	Id Artículo	Cantidad	Coste	Margen	Precio
1	2014-10-01	4	6	4	1	27,65 €	51,35 €	79,00 €
2	2014-10-01	3	6	7	1	69,98 €	129,97 €	199,95 €
3	2014-10-01	2	8	4	1	27,65 €	51,35 €	79,00 €
4	2014-10-01	4	1	1	1	52,47 €	97,44 €	149,90 €
5	2014-10-01	4	3	4	1	27,65 €	51,35 €	79,00 €
6	2014-10-01	2	4	1	1	52,47 €	97,44 €	149,90 €
7	2014-10-01	4	5	1	1	52,47 €	97,44 €	149,90 €
8	2014-10-01	3	1	4	1	27,65 €	51,35 €	79,00 €
9	2014-10-01	4	7	1	1	52,47 €	97,44 €	149,90 €
10	2014-10-01	1	5	4	1	27,65 €	51,35 €	79,00 €
...	...	...	...	...	...	...	...	...
15233	2017-03-31	2	8	9	1	69,98 €	10,00 €	79,98 €
15234	2017-03-31	3	7	3	1	43,75 €	6,25 €	50,00 €
15235	2017-03-31	2	10	6	1	24,48 €	3,50 €	27,98 €
15236	2017-03-31	1	7	3	1	43,75 €	6,25 €	50,00 €
15237	2017-03-31	1	5	6	1	24,48 €	3,50 €	27,98 €
15238	2017-03-31	3	2	6	1	24,48 €	3,50 €	27,98 €

Tabla 22. Histórico de pedidos. Modelo de predicción.

INGRESOS DIARIOS TOTALES



Figura 38. Ingresos totales diarios para las tres temporadas simuladas.

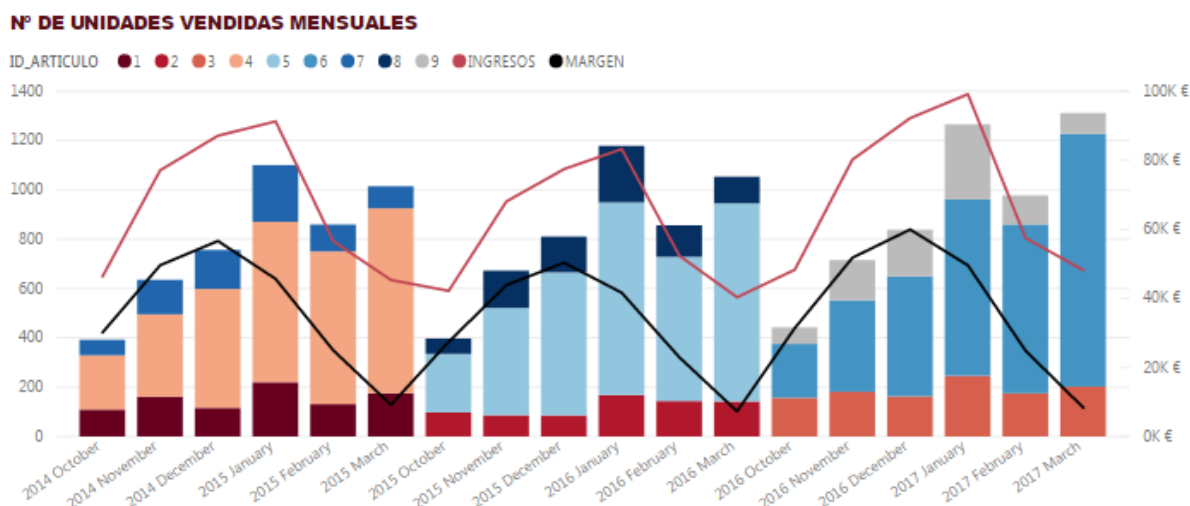


Figura 39. Análisis de las unidades vendidas mensualmente en todo el histórico.

En la Figura 38, se puede apreciar la estacionalidad existente en los ingresos diarios debido a los eventos y los períodos característicos de una temporada de invierno del sector. Con la generación de los pedidos se ha conseguido el objetivo de desarrollar un set de datos realista, que se ajusta a los parámetros deseados y, además, que contiene cierto ruido, que permite llevar a cabo un modelo de predicción que contiene todos los parámetros propios de un modelo del mundo de Data Science. Por otro lado, las transiciones que se muestran entre cada una de las temporadas han sido representadas de forma automática por el software empleado y no se tendrán en cuenta a la hora de ejecutar sobre estos datos el modelo de predicción.

Teniendo en cuenta las primeras impresiones que nos aporta la Figura 38, podemos llevar a cabo un análisis sobre los datos obtenidos mostrados en la Figura 39. En dicho gráfico se representan tanto las unidades vendidas para cada una de las referencias, como los ingresos y el margen sobre beneficios que implica la venta de cada una de las unidades.

Analizando en primer lugar las unidades vendidas, se puede ver como cumple con lo previsto ya que hay un claro aumento para la época de promociones y, sobre todo, en el mes de liquidación. A nivel económico, sin embargo, este mes de liquidación no aporta un buen margen ya que los precios son bastante bajos, siendo el coste de producción fijo. A parte, también era de esperar que el mes de mayor beneficio sea Diciembre ya que es el mes en el que sin haber promociones se producen un mayor número de ventas.

Finalmente, a nivel de artículos, las camisas son las más vendidas al ser una prenda estándar en la temporada de invierno y tener el precio más asequible. Sin embargo, la venta de chubasqueros y abrigos varia en función del clima en su mayor parte, aunque también por las promociones.

8.3. Modelos de predicción

Tras haber llevado a cabo la generación y correspondiente preparación de los datos que se van a emplear en el segundo modelo que comprende este proyecto, el siguiente paso será el estudio de los modelos de predicción que más se ajustan a las necesidades presentes y a la tipología de predicción que se quiere realizar. Por tanto, teniendo en cuenta que el objetivo es establecer una predicción de la demanda a partir de un histórico de datos de temporadas de invierno anteriores del negocio en cuestión, el tipo de modelo que más se ajusta son aquellos que forman parte de la rama de la tecnología de los datos denominada aprendizaje supervisado.

El aprendizaje supervisado es una técnica empleada en la minería de datos cuyo objetivo es el desarrollo de una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos. En otras palabras, el procedimiento a seguir comienza dividiendo el set inicial de datos en dos partes, un set de entrenamiento y uno de validación o testeo. Tomando los datos de entrenamiento, el modelo será capaz de generalizar a partir de los datos presentados a las situaciones no vistas previamente y, posteriormente, podrá comprobar la calidad de la predicción realizada con el set de validación.

A nivel general, el procedimiento básico del desarrollo de un modelo de aprendizaje supervisado sería el explicado anteriormente. Sin embargo, la metodología es algo más compleja ya que requiere, en primer lugar, realizar una comparativa previa de los posibles modelos a aplicar y, una vez elegido el mejor, optimizar dicho modelo a través del ajuste de sus parámetros intrínsecos con el fin de reducir los errores lo máximo posible y que, como consecuencia, la predicción buscada sea lo más precisa posible. El procedimiento mencionado se muestra en la Figura 40.

Este procedimiento está comprendido básicamente de cinco etapas dentro de un proceso iterativo:

1. Recolección de datos.
2. Separación del set de datos en entrenamiento y testeo.
3. Aprendizaje del algoritmo.
4. Evaluación.
5. Optimización o refinamiento.

En este proyecto, los dos modelos que se han seleccionado para desarrollar la predicción objetivo son un modelo ARIMA, que es un modelo estadístico basado en medias móviles. Y el segundo sería un modelo desarrollado por el equipo de Data Science de Facebook, cuyo principal fundamento son los modelos aditivos, es decir, modelos cambiantes que no siguen una tendencia lineal. Ambos modelos serán evaluados y optimizados, seleccionando el que más se adecúe a las necesidades del proyecto.

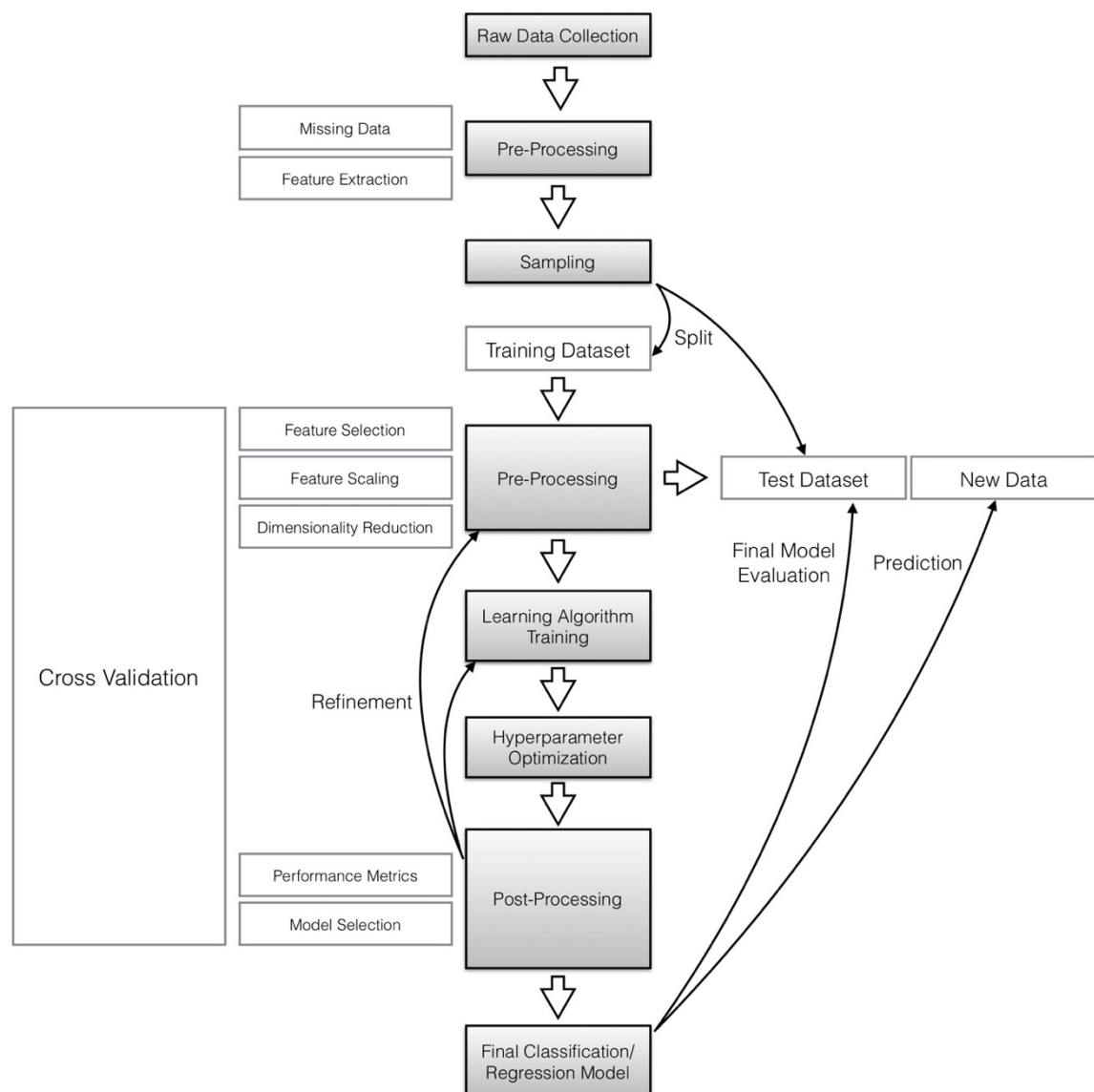


Figura 40. Diagrama de flujo del modelo general de aprendizaje supervisado. (Fuente: Sebastian Raschka, 2014).

8.3.1. Modelo ARIMA

El modelo ARIMA, modelo autorregresivo integrado de promedio móvil, es un modelo estadístico, aplicado fundamentalmente a series temporales, que emplea variaciones y regresiones de datos de tipo estadístico con el objetivo de encontrar patrones en dichos datos que permitan realizar una predicción hacia el futuro [BLEE17].

El modelo ARIMA requiere la identificación previa de una serie de coeficientes y números de regresiones que se utilizarán. La precisión con la que se determinen dichos coeficientes tendrá una gran influencia en los resultados finales, al ser muy sensible el modelo a este factor.

Suele expresarse como $ARIMA(p, d, q)$, donde los parámetros p, d, q son números enteros no negativos que indican el orden de las distintas componentes del modelo – respectivamente, las componentes autorregresiva, integrada y de media móvil.

En primer lugar, se procede a tomar la decisión en relación con la separación del set de datos generado. La idea es tener una referencia de datos notablemente buena y general con la cual poder comparar los resultados, es decir, que contenga datos de ventas de todo tipo de días, desde días con promociones a días normales del período de la temporada de invierno. Por lo tanto, se ha tomado de decisión de separar el set de datos para la posterior validación dejando los cuatro meses finales del histórico completo, o sea, de Diciembre de 2016 a Marzo de 2017. Con este set de datos se tienen tanto días con eventos festivos, promociones y días normales de venta. En resumen, los datos de entrenamiento comprenden un 86,75%, mientras que el set de testeo sería un 13,25%.

Cabe destacar que los datos de entrada al modelo $ARIMA$, a parte de los coeficientes p, d, q , son los datos de unidades totales vendidas para cada uno de los días del histórico de ventas generado. Es decir, el set de datos de entrada está comprendido por dos columnas, la primera sería la fecha día a día y la segunda sería el número de unidades vendidas para cada uno de los días.

Una vez realizada la separación de los datos de entrada al modelo, el siguiente paso a seguir es probar el modelo para confirmar que los resultados son aceptables. Por lo tanto, se decidió seleccionar los parámetros de $ARIMA$ de acuerdo con la experiencia de otros modelos similares para el sector. El modelo seleccionado es un modelo $ARIMA(5,1,1)$, es decir, contiene las tres componentes.

Con el fin de evaluar los resultados obtenidos, emplearemos dos indicadores diferentes. En primer lugar, analizaremos si cada término en el modelo es estadísticamente significativo y, posteriormente, observaremos tres variables de error propias de la minería de datos [HYND06], que son las siguientes:

- Error cuadrático medio (MSE o *Mean Square Error* en inglés). Mide el promedio de la diferencia entre el estimador y lo que se estima al cuadrado. Incorpora tanto la varianza del estimador así como su sesgo.
- Error absoluto medio (MAE o *Mean Absolute Error* en inglés). Es una medida propia de las series temporales que calcula, como bien dice su nombre, el error medio del valor absoluto de la diferencia entre el pronóstico y los resultados posibles.
- Error absoluto porcentual medio (MAPE o *Mean Absolute Percentage Error* en inglés). Determina el porcentaje de error de la predicción realizada sobre el valor real que se tiene para validar la medida obtenida.

Del modelo desarrollado se obtiene una tabla de resultados en la cuál se expresan una serie de parámetros estadísticos, los cuáles permiten la evaluación del pronóstico llevado a cabo.

ARIMA Model Results						
Dep. Variable:	D.y	No. Observations:	426			
Model:	ARIMA(5, 1, 1)	Log Likelihood	-1179.397			
Method:	css-mle	S.D. of innovations	3.831			
Date:	Tue, 19 Jun 2018	AIC	2374.794			
Time:	11:37:23	BIC	2407.229			
Sample:	1	HQIC	2387.606			
	coef	std err	z	P> z	[0.025	0.975]
const	-0.0017	0.002	-0.785	0.433	-0.006	0.002
ar.L1.D.y	-0.0063	0.048	-0.130	0.897	-0.101	0.088
ar.L2.D.y	0.0621	0.048	1.294	0.196	-0.032	0.156
ar.L3.D.y	0.0184	0.048	0.383	0.702	-0.076	0.113
ar.L4.D.y	0.0852	0.048	1.775	0.077	-0.009	0.179
ar.L5.D.y	0.1361	0.051	2.677	0.008	0.036	0.236
ma.L1.D.y	-1.0000	0.008	-131.982	0.000	-1.015	-0.985
Roots						
	Real	Imaginary	Modulus	Frequency		
AR.1	1.3369	-0.0000j	1.3369	-0.0000		
AR.2	0.3608	-1.4428j	1.4872	-0.2110		
AR.3	0.3608	+1.4428j	1.4872	0.2110		
AR.4	-1.3420	-0.8264j	1.5761	-0.4122		
AR.5	-1.3420	+0.8264j	1.5761	0.4122		
MA.1	1.0000	+0.0000j	1.0000	0.0000		

Figura 41. ARIMA Model Results. ARIMA(5, 1, 1).

En este resumen estadístico, la columna coef representa el peso de cada uno de los coeficientes que conforman el modelo ARIMA presentado sobre la serie temporal. Además, la columna $P > |z|$ provee información sobre la validez de cada uno de los coeficientes. Dichos coeficientes, para ser considerados aceptables, deben de tener un p-valor inferior o próximo al intervalo de confianza del modelo (95%, por defecto). Como se puede observar, la mitad de los valores son bastante superiores al valor buscado, por lo que se puede considerar que el modelo, con los parámetros seleccionados, es válido pero puede ser optimizado.

Además del resumen estadístico comentado, otros valores de salida que conforman el conjunto de indicadores estadísticos propios de los modelos de aprendizaje supervisado son los errores mencionados anteriormente, y que se muestran a continuación.

p	d	q	MAPE	MAE	MSE
5	1	1	5,63	18,18	51,71

Tabla 23. Errores estadísticos modelo ARIMA(5, 1, 1).

Tras haber comprobado la validez de esta técnica para el modelo en cuestión y viendo, según los valores mostrados en la tabla anterior, que los errores son aceptables, se procede a realizar un análisis de sensibilidad que permita optimizar los parámetros p , d y q con el fin de obtener mejores resultados. El análisis de sensibilidad es un proceso iterativo llevado a cabo dando valores a la función ARIMA que pueden dar lugar a modelos que no contengan todas las componentes de dicha técnica, es decir, la parte integral del modelo puede ser anulada si el coeficiente d es nulo, por ejemplo. Este hecho también permitirá ver el efecto que tiene cada una de las componentes de la función sobre los datos que se están tratando.

p	d	q	MAPE	MAE	MSE
3	0	1	5,60	18,18	51,14
3	0	2	5,60	18,18	51,14
3	1	1	5,60	18,18	51,14
3	1	2	5,60	18,18	51,14
5	0	1	5,64	18,18	51,70
5	0	2	5,63	18,18	51,56
5	1	1	5,63	18,18	51,71
5	1	2	5,64	18,18	51,70
7	0	1	5,63	18,18	51,56
7	0	2	5,61	18,18	51,35
7	1	1	5,63	18,18	51,64
7	1	2	5,63	18,18	51,68
10	0	1	5,64	18,18	51,72
10	0	2	5,59	19,01	50,87
10	1	1	5,64	18,18	51,74
10	1	2	5,64	18,18	51,81
12	0	1	5,61	17,36	51,58
12	0	2	5,62	18,18	51,55
12	1	1	5,64	17,36	51,82
12	1	2	5,64	18,18	51,81

Tabla 24. Análisis de sensibilidad. Modelo ARIMA.

Según los profesionales del sector y los resultados de otros casos de estudio relacionados con la materia tratada, el indicador que mejor determina la validez del modelo y la calidad de los resultados que genera es el Error Absoluto Porcentual Medio (MAPE). Por lo tanto, la elección de la combinación óptima de valores p , d y q se llevará a cabo en base al valor de este parámetro. De esta manera, se obtiene que el óptimo para este modelo es el formado por los siguientes valores: $p = 10$, $d = 0$ y $q = 2$.

Tal y como se ha mencionado antes, había la posibilidad de que alguno de los parámetros de la función ARIMA tomase el valor de cero, anulando la componente correspondiente. Por este motivo, el modelo óptimo para este proyecto sería un modelo ARMA en vez de ARIMA. A continuación, se presenta el resumen estadístico para el caso óptimo.

ARMA Model Results						
Dep. Variable:	y	No. Observations:	427			
Model:	ARMA(10, 2)	Log Likelihood	-1176.958			
Method:	css-mle	S.D. of innovations	3.794			
Date:	Tue, 19 Jun 2018	AIC	2381.915			
Time:	15:57:36	BIC	2438.710			
Sample:	0	HQIC	2404.348			
	coef	std err	z	P> z	[0.025	0.975]
const	0.6430	0.286	2.248	0.025	0.082	1.204
ar.L1.y	-1.0118	0.049	-20.600	0.000	-1.108	-0.916
ar.L2.y	-0.9376	0.069	-13.616	0.000	-1.073	-0.803
ar.L3.y	0.0622	0.082	0.755	0.451	-0.099	0.224
ar.L4.y	0.1500	0.082	1.830	0.068	-0.011	0.311
ar.L5.y	0.2257	0.083	2.716	0.007	0.063	0.389
ar.L6.y	0.2321	0.084	2.767	0.006	0.068	0.396
ar.L7.y	0.1803	0.084	2.140	0.033	0.015	0.345
ar.L8.y	0.0856	0.085	1.005	0.316	-0.081	0.253
ar.L9.y	0.0702	0.073	0.964	0.336	-0.072	0.213
ar.L10.y	0.0228	0.053	0.434	0.665	-0.080	0.126
ma.L1.y	1.0100	0.010	103.061	0.000	0.991	1.029
ma.L2.y	1.0000	0.011	94.530	0.000	0.979	1.021
Roots						
	Real	Imaginary	Modulus	Frequency		
AR.1	1.2249	-0.0000j	1.2249	-0.0000		
AR.2	-0.5143	-0.8628j	1.0045	-0.3356		
AR.3	-0.5143	+0.8628j	1.0045	0.3356		
AR.4	0.8919	-1.3226j	1.5953	-0.1556		
AR.5	0.8919	+1.3226j	1.5953	0.1556		
AR.6	0.0084	-1.5498j	1.5498	-0.2491		
AR.7	0.0084	+1.5498j	1.5498	0.2491		
AR.8	-1.3162	-0.8039j	1.5423	-0.4127		
AR.9	-1.3162	+0.8039j	1.5423	0.4127		
AR.10	-2.4391	-0.0000j	2.4391	-0.5000		
MA.1	-0.5050	-0.8631j	1.0000	-0.3343		
MA.2	-0.5050	+0.8631j	1.0000	0.3343		

Figura 42. ARMA Model Results. Modelo ARIMA óptimo.

Tal y como se comentó en el resumen del primer modelo simulado, los parámetros significativos son los coeficientes de la función ARIMA y sus correspondientes p-valor. Para este caso, se tiene que al haber 10 coeficientes en la función, estos toman valores próximos a 0,05 por lo que se puede decir que los resultados obtenidos son mejores que los previamente obtenidos, hecho que también se puede apreciar en el valor de cada uno de los errores en la Tabla 24.

Gráficamente, la predicción realizada y su correspondiente comparación con los valores reales de testeo se muestran en las figuras presentadas a continuación.



Figura 43. Predicción Modelo óptimo $ARIMA(10, 0, 2)$.

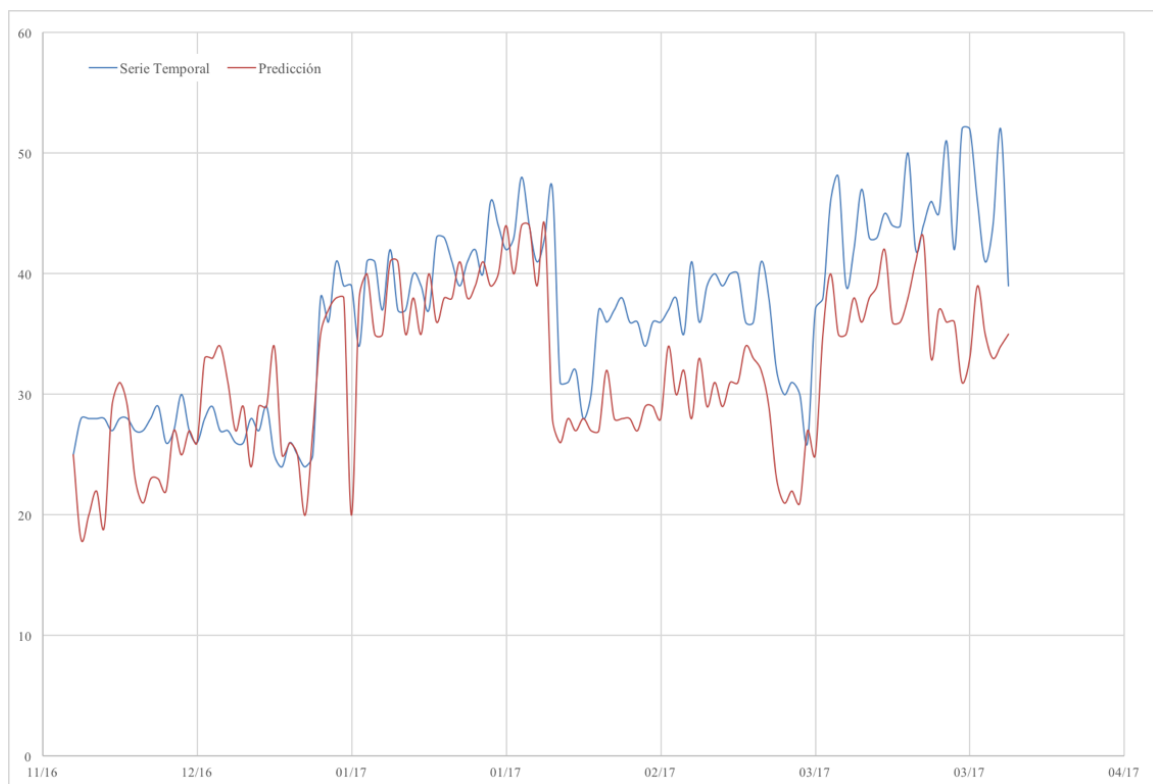


Figura 44. Comparativa de datos de testeo y predicción. $ARIMA(10, 0, 2)$.

8.3.2. Modelo FbProphet

El modelo desarrollado por Facebook, FbProphet, es un modelo de predicción de series temporales diseñado con el objetivo de poder ser aplicado a diversos aspectos implicados en el mundo de los negocios que sean necesarios de predecir. Es decir, el objetivo de los autores de dicho modelo fue el de diseñar un modelo de predicción generalista o, como ellos le llaman, predicción en escala [TAYL17].

Los dos primeros tipos de escala que tienen en cuenta es que este modelo sea adecuado para un gran número de usuarios, no necesariamente especializados en el mundo de la predicción y la minería de datos. Además, se pretende que el modelo pueda abarcar una gran cantidad de problemas que contengan factores potencialmente idiosincráticos. Por último, la tercera escala está relacionada con la facilidad del modelo para ser evaluado y ajustado cuando los resultados de la simulación sean pobres y el modelo no se esté comportando tal y como se espera.

El modelo está compuesto por una serie de parámetros intuitivos que puedan ser ajustados sin tener un conocimiento en detalle del desarrollo interno del modelo. Dichos parámetros son empleados para caracterizar las tres componentes principales que lo conforman: tendencia, estacionalidad y eventos. La tendencia modela los cambios no periódicos presentados en la serie temporal, mientras que la estacionalidad representa los cambios periódicos.

En el desarrollo de este modelo se ha empleado el mismo procedimiento que en el modelo ARIMA explicado en el apartado anterior. En primer lugar, la división de los datos de entrada será la misma que la usada previamente, es decir, un 87% aproximadamente de datos destinados al entrenamiento del modelo y un 13% de los datos para el posterior testeo.

El set de datos de entrada empleado en esta ocasión presenta ciertas variaciones con respecto al modelo de ARIMA debido a que para este caso se tienen en cuenta más aspectos que caracterizan al modelo en cuestión. Por una lado, se tiene la matriz de datos de entrada comprendida por una columna con la fecha de cada uno de los días que se van a proceder a simular y, por otro lado, el número de unidades vendidas en cada uno de los días, procedente de la generación de los datos llevaba a cabo con anterioridad. Además, las funciones del modelo reciben otra matriz de entrada donde se especifican los eventos significativos que condicionan el modelo.

- Black Friday.
- Festividades.
- Período de rebajas.
- Días de lluvia.
- Días con temperatura inferior a la temperatura media en la temporada de invierno.

Por otro lado, los parámetros de entrada que recibe la función principal del modelo son tres coeficientes que permiten caracterizar la escala de cada uno de los componentes del modelo: *seasonality_prior_scale*, *changepoint_prior_scale* y *holidays_prior_scale*. Estos tres parámetros toman, por defecto, unos valores previamente definidos por los autores del mismo y que son desconocidos. Sin embargo, el comportamiento por defecto del modelo frente a situaciones de negocio determinadas no tiene por que ser el óptimo siempre. Por esta razón, se ha llevado a cabo un análisis de sensibilidad que permite determinar qué valores son aquellos que permiten el mejor comportamiento para el caso de negocio en estudio. Para este modelo en cuestión, se comprobó a alto nivel que los resultados obtenidos con los valores por defecto eran buenos y se procedió directamente a la realización del análisis de sensibilidad.

A continuación, se presentan los resultados del análisis, en el que se evalúan los mismos errores estadísticos que en el modelo ARIMA y sobre los cuáles se determinará posteriormente qué modelo se tomará como definitivo para este proyecto.

Seasonality	Holidays	Changepoint	MAPE	MAE	MSE
2	0,5	0,05	5,54	14,08	48,22
2	0,5	0,1	5,48	13,91	47,41
2	0,5	0,5	4,20	10,82	30,49
2	1	0,05	5,53	14,03	48,17
2	1	0,1	5,45	13,84	47,04
2	1	0,5	4,21	10,85	30,62
2	5	0,05	5,55	14,11	48,36
2	5	0,1	5,49	13,94	47,52
2	5	0,5	4,10	10,62	29,31
5	0,5	0,05	5,53	14,04	48,10
5	0,5	0,1	5,46	13,87	47,26
5	0,5	0,5	4,19	10,80	30,34
5	1	0,05	5,54	14,07	48,25
5	1	0,1	5,46	13,87	47,15
5	1	0,5	4,12	10,66	29,51
5	5	0,05	5,53	14,04	48,18
5	5	0,1	5,48	13,91	47,42
5	5	0,5	4,17	10,76	30,13
10	0,5	0,05	5,52	14,03	48,12
10	0,5	0,1	5,46	13,86	47,15
10	0,5	0,5	4,17	10,76	30,13
10	1	0,05	5,53	14,04	48,08
10	1	0,1	5,45	13,85	47,12
10	1	0,5	4,13	10,68	29,65
10	5	0,05	5,55	14,09	48,36
10	5	0,1	5,45	13,85	47,09
10	5	0,5	4,18	10,78	30,22

Tabla 25. Análisis de sensibilidad. Modelo FbProphet.

El resultado del análisis de sensibilidad mostrado en la Tabla 25 indica que los valores que optimizan el modelo FbProphet para el caso de negocio que comprende este proyecto son los siguientes:

- *seasonality_prior_scale* = 2
- *changepoint_prior_scale* = 0,5
- *holidays_prior_scale* = 5

Al igual que en modelo ARIMA, el indicador empleado para determinar la combinación de valores óptima es el error MAPE. Un estudio realizado por cuatro investigadores de la Universidad de París en 2017 se lleva a cabo la demostración de por qué este error es un buen indicador para este tipo de modelos tanto a nivel teórico como práctico [MYTT17].

Por lo tanto, teniendo en cuenta los parámetros seleccionados para optimizar el modelo, las funciones propias del mismo llevan a cabo la simulación y correspondiente pronóstico del período de meses fijado, representado en la siguiente gráfica.

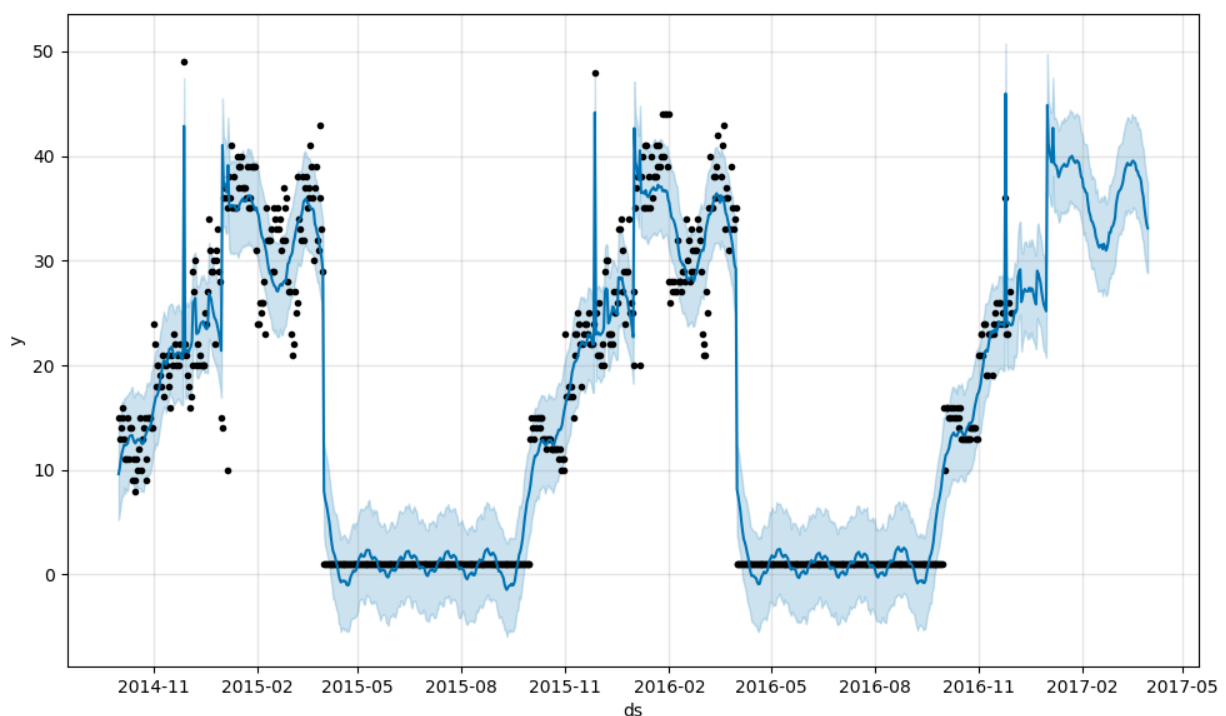


Figura 45. Predicción de la demanda de unidades. Modelo FbProphet.

En la Figura 45, hay que diferenciar tres series de datos principales. En primer lugar, la nube de puntos se corresponde con los datos de entrenamiento que se emplean como entrada al modelo. Por otro lado, la línea de color azul oscuro representa la línea de tendencia que tiene el pronóstico llevado a cabo por el modelo, mientras que la franja color celeste que acompaña a la línea de tendencia sería el intervalo de confianza que presentan los datos de salida.

De acuerdo con la relación gráfica existente entre la nube de puntos y la representación de la predicción llevada a cabo, se puede afirmar que el resultado obtenido es bueno. A parte, esta afirmación puede ser confirmada con los valores de los errores mostrados en negrita en la Tabla 25, que hacen referencia a este último caso simulado.

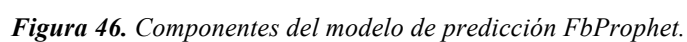
Las funciones del modelo, a parte de la gráfica mostrada anteriormente, también generan un conjunto de gráficas que representan cada una de las componentes principales involucradas en el modelo, como son la estacionalidad, los eventos y la tendencia. Esta gráfica se muestra a continuación en la Figura 46.

En la primera gráfica se muestra la tendencia que tiene la serie temporal a lo largo del período de tiempo simulado. Hablando de forma general, la tendencia tiende a ser creciente, sobre todo en la última temporada, pese al ruido existente en la primera mitad. Teniendo en cuenta la tendencia final, todo indica que de cara a realizar un predicción futura el nivel de ventas del negocio objeto del proyecto aumentará.

En segundo lugar, se representan los eventos considerados a la hora de modelar cada una de las temporadas de invierno que conforman el histórico. A primera vista, se aprecia como los días de Black Friday están perfectamente representados. Además, los períodos de vacaciones de Navidad y las posteriores rebajas que lo suceden año a año también están definidos correctamente. El impacto de estos eventos sobre la serie temporal es importante debido a que año a año se repiten, de ahí la relevancia de incorporarlos al modelo.

Por último, las últimas dos gráficas representan el concepto de la estacionalidad sobre el modelo a nivel anual y semanal. Teniendo en cuenta el set de datos de entrada aportado como entrada, las funciones son capaces de identificar el patrón que rige el comportamiento del negocio de forma anual, en el cuál se representa perfectamente el incremento de las ventas a partir de Noviembre hasta Enero incluido, además del pico generado en el mes de Marzo provocado por las liquidaciones. Sin embargo, durante la temporada de verano no hay actividad ya que no es objeto del proyecto.

Por otro lado, a nivel semanal las ventas aumentan a partir del jueves hasta el domingo, aunque los lunes toman el valor más alto a causa de las fechas de los días festivos y de las ventas en rebajas.



8.4. Resultados

Teniendo en cuenta los resultados obtenidos en los dos modelos desarrollados anteriormente, la parte final de la metodología de predicción consistiría en elegir aquel que mejor se ajusta a las necesidades y objetivos fijados para el caso de negocio objeto de este proyecto. Para ello, tal y como se ha comentado en los apartados en los que se ha desarrollado cada uno, los principales indicadores de validez en análisis de series temporales son los errores estadísticos, a parte de los indicadores y resúmenes estadísticos propios de cada modelo.

Por lo tanto, se exponen a continuación los valores que dichas magnitudes toman para los casos óptimos desarrollados en cada uno de los modelos.

Modelo	MAPE	MAE	MSE
ARIMA	5,59	19,01	50,87
FbProphet	4,10	10,62	29,31

Tabla 26. Análisis comparativo de los modelos de predicción.

A la vista de los resultados obtenidos, se puede afirmar con claridad que el modelo que mejor respuesta tiene es el desarrollado por Facebook, FbProphet. Los tres errores toman valores bastante inferiores con respecto a los del modelo ARIMA, lo cual significa que el ajuste de la línea de predicción es más preciso para el caso del modelo desarrollado por Facebook.

A parte de los errores, hay una serie de motivos por los cuales se ha tomado la decisión de que el modelo FbProphet es el más adecuado para el caso de estudio. Se considera, primeramente, que este modelo es más completo debido a que tiene en cuenta una serie de aspectos que permiten añadir los dos factores que desde el principio eran objetivo del proyecto, el clima y las promociones. Además, no sólo permite incorporarlos, sino que también los incorpora y trata de forma correcta proveyendo una serie de resultados adecuados para el nivel de precisión pretendido en la realización de un pronóstico de demanda.

Aunque el modelo ARIMA es considerado uno de los más completos en cuanto a regresión lineal se refiere, con la implementación de FbProphet se han obtenido una serie de ventajas respecto a ARIMA:

- Flexibilidad. Es más fácil tratar la estacionalidad con múltiples períodos y que el usuario haga sus propias hipótesis sobre la tendencia.
- A diferencia de ARIMA, las medidas no tienen por qué estar regularmente espaciadas, no teniendo que interpolar para obtener valores ausentes.
- Es rápido a la hora de ser ajustado.
- Está compuesto por una serie de parámetros fácilmente interpretables que pueden ser

modificados por el usuario de cara a ajustar el modelo a sus propias necesidades.

En resumen, el modelo FbProphet es más adecuado para el desarrollo de una predicción de la demanda de las características fijadas en este proyecto no sólo por los resultados estadísticos obtenidos, sino también por las posibilidades y amplio rango de características que permite implementar.

9. Conclusiones

Tras la realización de este proyecto, se ha observa la necesidad imperiosa que tienen las empresas del sector por instaurar este tipo de herramientas en su día a día. Pese a que suponen una fuerte inversión inicial y la contratación de profesionales especializados en sector, la repercusión tanto a nivel económico como de gestión resulta ser muy significativa.

Es interesante mencionar el concepto del que tanto se habla en la actualidad, gestión del cambio. El sector retail ha experimentado este proceso en los últimos años de forma precipitada y, por este motivo, las empresas del sector han tenido que reaccionar hábilmente ante tal situación. Aunque el avance vertiginoso de la tecnología ha propiciado este hecho, a su vez, también permite a los profesionales desarrollar nuevas e innovadoras soluciones con las que adaptarse a las nuevas tendencias y poder convertirse en fuertes competidores del mercado.

Los sistemas de gestión de pedidos son un claro ejemplo de estas herramientas ya que, como se ha demostrado en el desarrollo del modelo, tienen efecto sobre diversos procesos dentro de la cadena de suministro. Por un lado, permite llevar a cabo un estudio de evaluación y posible reestructuración de la red de distribución de los negocios pero también es útil a la hora de establecer nuevas reglas y estrategias en el proceso de abastecimiento del inventario.

Todas estas mejoras tienen un fuerte efecto sobre los costes pero también afectan considerablemente a la operativa de la empresa y a sus resultados a nivel de eficiencia. Claros ejemplos son las empresas líderes del sector, como Amazon o Alibaba [WANG17].

Por otro lado, el desarrollo de un modelo de predicción de demanda permite a las empresas tener un conocimiento previo del comportamiento del mercado y de sus clientes frente a las fluctuaciones propias de cada uno de ellos. Ser conocedor de esta información aporta una ventaja de cara al futuro proceso de abastecimiento y de fijación de precios con el fin de obtener el mayor margen de beneficios posible.

El modelo de predicción es un complemento perfecto para el sistema de optimización OMS desarrollado ya que facilita a las empresas los procesos de reestructuración de su red de distribución y la gestión eficiente de su inventario y posterior cobertura de demanda. Por tanto, como consecuencia de la implantación de estos sistemas lo que se obtiene es un ahorro de costes a la vez que se mejora en eficiencia y, por lo tanto, en la imagen dada a los clientes.

En relación con esto, es interesante mencionar el interés de las empresas por anteponerse al mercado, pudiendo lanzar nuevas campañas de promoción fuera de los períodos normalmente establecidos o, también, a nivel interno conocer la repercusión de las nuevas modas y tendencias que caracterizan el sector. En el modelo actual de mercado, el desarrollo de este tipo de estrategias son las que hacen que las empresas destaquen sobre las demás ya que el acceso a la información y a la nueva tecnología es algo que cada vez supone menor dificultad para las empresas.

10. Bibliografía

- [AMAZ16] Amazon (Dirección). (2016). Amazon Prime Air [Película].
- [BROW16] Brown, S. (2016). The Second Era of Digital Retail. Intel. The Store WPP.
- [BLEE17] Brownlee, J. (2017). Machine Learning Mastery. Obtenido de <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>
- [CABI17] Cabify. (05 de 2017). Inside Cabify. Obtenido de <https://inside.cabify.com/colombia/llego-cabify-express>
- [CHOP13] Chopra, S., & Meindl, P. (2013). Supply Chain Management. Strategy, Planning and Operation. Pearson.
- [DOUG14] Douglas, M. (Agosto de 2014). New Retail Strategies: It's a Store! It's a Site! It's a Warehouse! Obtenido de InboundLogistics: <http://www.inboundlogistics.com/cms/article/new-retail-strategies-its-a-store-its-a-site-its-a-warehouse/>
- [ESTR07] Estrada Romeu, M. À. (2007). Análisis de estrategias eficientes en la logística de distribución de paquetería.
- [GIBS14] Gibson, B. J., & Defee, C. C. (2014). The State of the Retail Supply Chain. JDA.
- [HABI08] Habib, M., & Jungthirapanich, C. (2008). An integrated framework for research and education supply chain for the universities.
- [HODS17] Hodson, N., Perrigo, C., & Hardman, D. (2017). PwC. Obtenido de Strategy&: <https://www.strategyand.pwc.com/trend/2017-retail-trends>
- [HYND06] Hyndman, R. J. (2006). Another look at forecast-accuracy metrics for intermittent demand. Monash University, Statistics.
- [MERI17] Merino, P. P. (3 de 3 de 2017). E-commercenews. Obtenido de E-commercenews: <https://ecommerce-news.es/juan-roig-mercadona-nuestra-web-actual-una-mierda-56155>
- [MOBI16] Mobile Marketing Association. (2016). Mobile Marketing Association. Obtenido de Case Studies: http://www.mmaglobal.com/case-study-hub/case_studies/view/31812
- [MYTT17] Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2017). Mean Absolute Percentage Error for Regressions Models.
- [RAO_16] Rao, D. S. (2016). Machine Learning Explained. PwC.

- [SERR17] Serracanta, M. (19 de Enero de 2017). Tendencias en Supply Chain: ¿aceleración imparable? Obtenido de EAE Business School:
<https://www.eae.es/actualidad/faculty-research/tendencias-en-supply-chain-aceleracion-imparable>
- [SIDE17] Sides, R., Pingitore, G., & Ament, L. (2017). 2017 retail, wholesale, and distribution industry outlook. Deloitte, Industry Insights.
- [TAYL17] Taylor, S. J., & Letham, B. (2017). Forecasting at Scale. Facebook.
- [TOMP09] Tompkins Associates. (2009). Effectiveness, 15 Key Factors That Impact Your Distribution Network. Obtenido de Supply Chain Edge:
<https://www.tompkinsinc.com/publications/competitive>
- [TORR17] Torres, C. (2017). A República. Obtenido de A fin de mes:
<http://afindemes.republica.com/finanzas/inteligencia-artificial-en-futuro-para-tus-inversiones.html>
- [VISS16] Visser, J., & Uiterweerd, M. (2016). Capgemini Consulting. Obtenido de <https://www.capgemini.com/consulting/2016/02/fulfilment-in-omnichannel-retailing-the-key-trends-to-return-to-growth/>
- [WANG17] Wang, H. H. (12 de 11 de 2017). Forbes. Obtenido de Forbes:
<https://www.forbes.com/sites/helenwang/2017/11/12/alibabas-singles-day-by-the-numbers-a-record-25-billion-haul/#5f0ebe141db1>
- [WILL15] Williams, B. (2015). Supply Chain Management Consulting in Europe. Kennedy Consulting Research & Advisory. ALM Media Properties.

11. Anexos

11.1. Anexo 1: Generador de Pedidos


```
#Generador de pedidos

import pandas as pd
import numpy as np
import random
from datetime import datetime, timedelta

articulos = pd.read_excel('Articulos.xlsx')
clientes = pd.read_excel('Clientes.xlsx')
tiendas = pd.read_excel('Tiendas.xlsx')
inputs = pd.read_excel('Inputs.xlsx', index_col = 0)

#Inputs
#Para que funciones este script tiene que tener en el excel de inputs Los datos de Los pedidos por dia min y max
mes_inicio = inputs['INPUTS']['Mes_inicio']
mes_fin = inputs['INPUTS']['Mes_fin']

#Inicializacion
cont = 0
contador = 1
data = {'ID_PEDIDO': [], 'ID_LINEA_PEDIDO': [], 'FECHA_PEDIDO': [], 'ID_TIENDA': [], 'ID_CLIENTE': [], 'ID_ARTICULO': [], ...
        ...'CANTIDAD': [], 'PRECIO': []}
cols = ['ID_PEDIDO', 'ID_LINEA_PEDIDO', 'FECHA_PEDIDO', 'ID_TIENDA', 'ID_CLIENTE', 'ID_ARTICULO', 'CANTIDAD', 'PRECIO']
art_gen = []

ventas_totales_dia = int(tiendas['VENTAS_TOTALES_MENSUAL_TIENDA'].sum()/30)

pedidos = pd.DataFrame(columns = cols)

np.random.seed(0)

for mes in range(int(mes_inicio),int(mes_fin)):
    dia = 1
    while dia <= 30:
        aux = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
        aux1 = tiendas['VENTAS_TOTALES_MENSUAL_TIENDA']/30
        aux2 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
        respuesta = True
        for index in range (0, len(aux)-1):

            while respuesta == True:

                num_ped_cli = 1

                #Indice del cliente
                indice_cli = 0
                cli_ind = np.random.randint(0, len(clientes)-1)
                data['ID_PEDIDO'] = contador

                #Generar fecha del pedido
                hora = np.random.randint(0,23)
                fecha = str(datetime(2017, mes, dia, hora, np.random.randint(0,59), np.random.randint(0,59)))
                data['FECHA_PEDIDO'] = fecha

                #Seleccion de la familia de articulo por genero
                if clientes['GENERO'][cli_ind] == 'M':
                    fam_art = random.randrange(1,4,2)
                else:
                    fam_art = random.randrange(1,3,1)

                art_gen = articulos[articulos['ID_FAMILIA'] == fam_art]

                #Seleccion de tienda
                id_t = 15 #ID de la tienda online por defecto
                for lab, row in tiendas.iterrows():
                    if clientes['LATITUD'][cli_ind] == tiendas['LATITUD'][lab]:
                        id_t = tiendas['ID_TIENDA'][lab]
                data['ID_TIENDA'] = tiendas['ID_TIENDA'][id_t-1]

                if aux[id_t-1] < aux1[id_t-1]:
                    contador += 1

                #Seleccion de articulos por cliente (lineas de pedido)
                while num_ped_cli <= int(clientes['UPT'][cli_ind]):

                    indice_cli = cli_ind
                    #Completar en DataFrame articulo

                    data['ID_LINEA_PEDIDO'] = num_ped_cli
                    data['ID_CLIENTE'] = clientes['ID_CLIENTE'][indice_cli]
```

```

#Indice del articulo
art_ind = random.randrange(min(art_gen['ID_ARTICULO'].index),max(art_gen['ID_ARTICULO'].index))
data['ID_ARTICULO'] = art_gen['ID_ARTICULO'][art_ind]

data['CANTIDAD'] = 1
data['PRECIO'] = art_gen['PRECIO'][art_ind]

if aux[id_t-1] < aux1[id_t-1]:
    aux[id_t-1] += art_gen['PRECIO'][art_ind]

    #Escribir en dataframe
    pedidos.loc[cont, cols] = data

cont += 1
num_ped_cli += 1

if aux[id_t-1] > aux1[id_t-1]:
    aux2[id_t-1] = 1

if sum(aux2) == len(aux2):
    respuesta = False

print("Generando pedidos del dia:" + str(dia) + " y mes: " + str(mes))
dia += 1

pedidos = pedidos.reset_index()
del pedidos['index']

#print(pedidos)
pedidos.to_excel('Pedidos.xlsx', index = False)

```

11.2. Anexo 2: Generador de Devoluciones


```

#Generador de devoluciones

import pandas as pd
import numpy as np
import random
from datetime import datetime, timedelta

tiendas = pd.read_excel('Tiendas.xlsx')
pedidos = pd.read_excel('Pedidos.xlsx')

#Generacion de devoluciones
#Preparacion de Los datos
ped_tiemdas = pedidos.sort_values(['ID_TIENDA'])
ped_tiemdas = ped_tiemdas.reset_index()
del ped_tiemdas['index']

#Seleccion de Los pedidos devueltos
porc_dev = tiendas['PORCENTAJE_DEVOLUCIONES'] #Porcentaje de devoluciones fijado en el maestro de tiendas

#Definicion dataframe devoluciones
cols_dev = ['ID_DEVOLUCION', 'ID_PEDIDO', 'FECHA_PEDIDO', 'FECHA_DEVOLUCION', 'ID_TIENDA', 'ID_CLIENTE', 'ID_ARTICULO',...
            ...'CANTIDAD', 'PRECIO']
dev = {'ID_DEVOLUCION': [], 'ID_PEDIDO': [], 'FECHA_PEDIDO': [], 'FECHA_DEVOLUCION': [], 'ID_TIENDA': [], 'ID_CLIENTE': [],...
        ...'ID_ARTICULO': [], 'CANTIDAD': [], 'PRECIO': []}
devoluciones = pd.DataFrame(columns = cols_dev)

cont_dev = 1
np.random.seed(0)

#Generacion
for id_t in range(1,len(tiendas)+1):
    num_dev = 1
    id_dev = 0
    while num_dev <= len(ped_tiemdas[ped_tiemdas['ID_TIENDA'] == id_t]) * porc_dev[id_t-1]:
        id_dev = np.random.randint(min(ped_tiemdas[ped_tiemdas['ID_TIENDA'] == id_t].index), max(ped_tiemdas[ped_tiemdas['ID_TIENDA']...
            ...== id_t].index))
        dev['ID_PEDIDO'] = ped_tiemdas['ID_PEDIDO'][id_dev]
        dev['FECHA_PEDIDO'] = ped_tiemdas['FECHA_PEDIDO'][id_dev]
        fecha_dev = datetime.strptime(ped_tiemdas['FECHA_PEDIDO'][id_dev], "%Y-%m-%d %H:%M:%S") + timedelta(days=1)
        dev['FECHA_DEVOLUCION'] = fecha_dev
        dev['ID_TIENDA'] = ped_tiemdas['ID_TIENDA'][id_dev]
        dev['ID_CLIENTE'] = ped_tiemdas['ID_CLIENTE'][id_dev]
        dev['ID_ARTICULO'] = ped_tiemdas['ID_ARTICULO'][id_dev]
        dev['CANTIDAD'] = ped_tiemdas['CANTIDAD'][id_dev]
        dev['PRECIO'] = ped_tiemdas['PRECIO'][id_dev]

        devoluciones.loc[cont_dev, cols_dev] = dev

        num_dev += 1
        cont_dev += 1
    print('Devoluciones generadas para la tienda: ' + str(id_t))

devoluciones = devoluciones.drop_duplicates(subset = 'ID_PEDIDO', keep = 'first', inplace = False)

devoluciones['FECHA_PEDIDO'] = pd.to_datetime(devoluciones['FECHA_PEDIDO'])
devoluciones = devoluciones.sort_values(['FECHA_PEDIDO'])

devoluciones = devoluciones.reset_index()
del devoluciones['index']

for ind in range(0, len(devoluciones)):
    dev['ID_DEVOLUCION'] = ind + 1
    devoluciones.loc[ind, 'ID_DEVOLUCION'] = dev['ID_DEVOLUCION']

#print(devoluciones)
devoluciones.to_excel('Devoluciones.xlsx', index = False)

```


11.3. Anexo 3: Generador de Stock Inicial


```
#Generador de stock
```

```
import pandas
as pd import
numpy as np
import random
from datetime import datetime, timedelta

inputs = pd.read_excel('Inputs.xlsx',
index_col = 0) if inputs['INPUTS']['Caso_1']
== 'Si':
    tiendas =
pd.read_excel('Tiendas_C1.xlsx') if
inputs['INPUTS']['Caso_2'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C2.xlsx')
```

```
pedidos =
pd.read_excel('Pedidos.xlsx')
articulos =
pd.read_excel('Articulos.xlsx')
pedidos =
pd.read_excel('Pedidos.xlsx')
```

```
#Preparacion de los datos
```

```
ped_tiendas =
pedidos.sort_values(['ID_TIENDA'])
ped_tiendas = ped_tiendas.reset_index()
del ped_tiendas['index']
```

```
#Generador de stock inicial
```

```
stock_0_i = pd.DataFrame(columns = ['ID_TIENDA', 'ID_ARTICULO', 'CANTIDAD',
'PRECIO', 'CANTIDAD_MONTO'])

id_art = articulos['ID_ARTICULO'] cont_i = 1
num_t = 1
for num_t in range(1,len(tiendas)+1): num_a = 1
    while num_a <= len(id_art):
        stock_0_i.loc[cont_i, 'ID_TIENDA'] = num_t
        stock_0_i.loc[cont_i, 'ID_ARTICULO'] = num_a
        stock_0_i.loc[cont_i, 'CANTIDAD'] = 0
        stock_0_i.loc[cont_i, 'PRECIO'] =
float(articulos[articulos['ID_ARTICULO'] == num_a]['PRECIO'].values)
        stock_0_i.loc[cont_i, 'CANTIDAD_MONTO'] = 0
        num_a += 1
        cont_i += 1

for id_tienda in range(1,len(tiendas)+1):
    for a in range(min(stock_0_i[stock_0_i['ID_TIENDA'] == id_tienda].index),
max(stock_0_i[stock_0_i['ID_TIENDA'] == id_tienda].index)):

        for num_p in range(min(ped_tiendas[ped_tiendas['ID_TIENDA'] ==
id_tienda].index), max(ped_tiendas[ped_tiendas['ID_TIENDA']...
...== id_tienda].index)):
            if ped_tiendas['ID_ARTICULO'][num_p] ==
stock_0_i['ID_ARTICULO'][a]:
                stock_0_i.loc[a, 'CANTIDAD'] += 1
```

```

stock_0_i.loc[a, 'CANTIDAD_MONTO'] += ped_tiendas['PRECIO'][num_p]

print('Stock inicial generado para la tienda: ' + str(id_tienda))

stock_0_i.to_excel('Stock_Inicial.xlsx', index = False)
#Modificador de stock para Escenarios 2, 3 y 4

inputs = pd.read_excel('Inputs.xlsx',
index_col = 0) if inputs['INPUTS']['Caso_1']
== 'Si':
    tiendas =
pd.read_excel('Tiendas_C1.xlsx') if
inputs['INPUTS']['Caso_2'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C2.xlsx')

stock_0_i =
pd.read_excel('Stock_Inicial.xlsx') porc_stock
= tiendas['PORCENTAJE_STOCK']

porc_stock = tiendas['PORCENTAJE_STOCK']

stock_0 = stock_0_i.copy()

for t in range(1,len(tiendas)+1):
    p_stock_t = porc_stock[t-1]

    #Selección del número de artículos a modificar
    if p_stock_t < 1:
        num_a = (1 - p_stock_t) * sum(stock_0_i[stock_0_i['ID_TIENDA'] ==
...t]['CANTIDAD']) elif p_stock_t > 1:
        num_a = (p_stock_t - 1) * sum(stock_0_i[stock_0_i['ID_TIENDA'] ==
t]['CANTIDAD']) elif p_stock_t == 1:
        num_a = p_stock_t * len(stock_0_i[stock_0_i['ID_TIENDA'] == t])

n = 0
respuesta = True
while respuesta == True:
    for n_a in range(min(stock_0[stock_0['ID_TIENDA'] == t].index),
max(stock_0[stock_0['ID_TIENDA'] == t].index)+1): if n <= num_a and
stock_0['CANTIDAD'][n_a] > 0:

        if p_stock_t < 1:
            v_cant = stock_0.loc[n_a, 'CANTIDAD'] - 1
            stock_0.loc.__setitem__((n_a, 'CANTIDAD'),
v_cant)
            v_pre = stock_0.loc[n_a, 'CANTIDAD_MONTO'] -
stock_0_i['PRECIO'][n_a] stock_0.loc.__setitem
((n_a, 'CANTIDAD_MONTO'), v_pre)
        elif p_stock_t > 1:
            v_cant =
stock_0.loc[n_a, 'CANTIDAD
'] + 1 stock_0.loc.
setitem
((n_a, 'CANTIDAD'),
v_cant)

```

```

        v_pre = stock_0.loc[n_a, 'CANTIDAD_MONTO'] +
        stock_0_i['PRECIO'][n_a] stock_0.loc.__setitem
        ((n_a, 'CANTIDAD_MONTO'), v_pre)

    n += 1

    if n >= num_a:
        respuesta = False

    print('Modificando stock de la tienda: ' + str(t))

stock = stock_0

if inputs['INPUTS']['Caso_1'] == 'Si':
    stock.to_excel('Stock_E1_C1.xlsx', index = False)
if inputs['INPUTS']['Caso_2'] == 'Si':
    stock.to_excel('Stock_E1_C2.xlsx', index =
    False)

#Distribuidor de stock

articulos = pd.read_excel('Articulos.xlsx')
inputs = pd.read_excel('Inputs.xlsx', index_col = 0)

if inputs['INPUTS']['Caso_1'] == 'Si':
    tiendas =
    pd.read_excel('Tiendas_C1.xlsx')
    st_0 =
    pd.read_excel('Stock_E1_C1.xlsx')
if inputs['INPUTS']['Caso_2'] == 'Si':
    tiendas =
    pd.read_excel('Tiendas_C2.xlsx') st_0 =
    pd.read_excel('Stock_E1_C2.xlsx')

st_0 = st_0.copy()

#Ajuste del stock del almacen entre tiendas
p_almacen = inputs['INPUTS']['Stock_almacen'] #porcentaje de pedidos que va
al stock del almacen
p_tiendas = 1 - p_almacen #porcentaje de pedidos que va al stock de las
tiendas

ud_almacen = int(sum(st_0[st_0['ID_TIENDA'] == 15]['CANTIDAD']) *
p_almacen) ud_tiendas = int(sum(st_0[st_0['ID_TIENDA'] == 15]['CANTIDAD'])
* p_tiendas) ud_tienda = int(ud_tiendas / (len(tiendas) - 1))

aux = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
aux1 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
respuesta = True np.random.seed(0)

print('Redistribuyendo stock de almacen por
tiendas...') for t in range(0, len(aux)-1):

    while respuesta == True:

```

```

for s in range(min(st_0[st_0['ID_TIENDA'] == 15].index),
               max(st_0[st_0['ID_TIENDA'] == 15].index)): id_t =
    np.random.randint(0, len(tiendas)-1)

    if st_0.loc[s, 'CANTIDAD'] > 0:

        for a in range(min(st_0[st_0['ID_TIENDA'] == id_t+1].index),
                       max(st_0[st_0['ID_TIENDA'] == id_t+1].index)): id_a =
            st_0.loc[a, 'ID_ARTICULO']
            if st_0.loc[a, 'ID_ARTICULO'] == st_0.loc[s, 'ID_ARTICULO']:
                valorc_t = st_0.loc[a, 'CANTIDAD'] + 1
                st_0.loc.__setitem__((a, 'CANTIDAD'), valorc_t)
                valorp_t = st_0.loc[a, 'CANTIDAD_MONTO'] +
                    articulos.loc[id_a, 'PRECIO'] st_0.loc.__setitem...
                    ...((a, 'CANTIDAD_MONTO'), valorp_t)

            valorc_c = st_0.loc[s, 'CANTIDAD'] - 1 st_0.loc.__setitem...
            ...((s, 'CANTIDAD'), valorc_c)
            valorp_c = st_0.loc[s, 'CANTIDAD_MONTO'] - st_0.loc[s, 'PRECIO']
            st_0.loc.__setitem__((s, 'CANTIDAD_MONTO'), valorp_c)
            aux[id_t - 1] += 1

    else:
        s += 1

    if aux[id_t - 1] > ud_tienda:
        aux1[id_t - 1] = 1

    if sum(aux1) ==
        len(aux1):
        respuesta = False

if inputs['INPUTS']['Caso_1'] == 'Si':
    st_0.to_excel('Stock_E2_C1.xlsx', index = False)

if inputs['INPUTS']['Caso_2'] == 'Si': st_0.to_excel('Stock_E2_C2.xlsx', index =
False)

```

11.4. Anexo 4: OMS. Escenario 1.

#OMS. Escenario 1

```
import pandas as pd
import numpy as np

#Cargar maestros y BBDDs
pedidos = pd.read_excel('Pedidos.xlsx')
clientes = pd.read_excel('Clientes.xlsx', index_col = 0)
inputs = pd.read_excel('Inputs.xlsx', index_col = 0)

if inputs['INPUTS']['Caso_1'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C1.xlsx')
    st_0 = pd.read_excel('Stock_E1_C1.xlsx')
if inputs['INPUTS']['Caso_2'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C2.xlsx')
    st_0 = pd.read_excel('Stock_E1_C2.xlsx')

centros = pd.read_excel('Centros.xlsx')
articulos = pd.read_excel('Articulos.xlsx')
devoluciones = pd.read_excel('Devoluciones.xlsx')
distancias = pd.read_excel('Distancias.xlsx', index_col = 0)
costes = pd.read_excel('Costes.xlsx', index_col = 0)

st_0 = st_0.copy()

#Costes
c_p_t = costes['COSTE']['Coste_preparacion_t']
c_p_a = costes['COSTE']['Coste_preparacion_a']
c_t = costes['COSTE']['Coste_transporte']
c_ep = costes['COSTE']['Coste_envio_p']
c_em = costes['COSTE']['Coste_envio_m']
c_eg = costes['COSTE']['Coste_envio_g']

#Definicion de dataframes
cols_d = ['ID_PEDIDO', 'FECHA_PEDIDO', 'DISPONIBILIDAD', 'ORIGEN', 'LAT_ORIGEN', 'LONG_ORIGEN', 'DESTINO',
          'LAT_DESTINO', 'LONG_DESTINO', 'ID_CLIENTE', 'ID_ARTICULO', 'CANTIDAD', 'PRECIO']
distribucion = pd.DataFrame(columns = cols_d) #DF de asignacion de origenes de pedidos
outputs = pd.DataFrame(columns = ['Valor', 'Unidades', 'Porcentaje']) #DF de outputs

#Definicion de outputs
cont = 0
coste_prep_t = 0
coste_prep_a = 0
coste_transp_t = 0
coste_transp_a = 0
coste_env_t = 0
coste_env_a = 0
coste_rot_t = 0
coste_rot_a = 0
coste_total = 0
n_rot_stock_t = 0
n_rot_stock_a = 0

#Filtrar dataframes
tiendas_by_1 = tiendas.set_index('ID_TIENDA')
tiendas_by_2 = tiendas.set_index('TIENDA')
centros_by_1 = centros.set_index('CENTROS')
articulos_by_1 = articulos.set_index('ID_ARTICULO')

#dividir las distancias en tiendas y centros
dist_c = distancias.loc['Almacen',:]
dist_t = distancias.loc['Tienda 1',:]

i = 0 #Contador para el DF
for id_p in range(0, len(pedidos)):
    distribucion.loc[i, 'ID_PEDIDO'] = pedidos['ID_PEDIDO'][id_p]
    distribucion.loc[i, 'FECHA_PEDIDO'] = pedidos['FECHA_PEDIDO'][id_p]
    distribucion.loc[i, 'ID_CLIENTE'] = pedidos['ID_CLIENTE'][id_p]
    distribucion.loc[i, 'ID_ARTICULO'] = pedidos['ID_ARTICULO'][id_p]

    dist_t_min = dist_t[pedidos['ID_CLIENTE'][id_p]].sort_values(ascending = True)
    #Ordena las tiendas por cercania segun el pedido

    #Seleccion de los datos de origen y destino en funcion de la tienda
```

```

if pedidos['ID_TIENDA'][id_p] != 15:

    #Origen
    distribucion.loc[i, 'ORIGEN'] = distancias.loc[:,pedidos['ID_CLIENTE'][id_p]].idxmin()
    distribucion.loc[i, 'LAT_ORIGEN'] = tiendas_by_2.loc[distancias.loc[:,pedidos['ID_CLIENTE']
    [id_p]].idxmin(), 'LATITUD']
    distribucion.loc[i, 'LONG_ORIGEN'] = tiendas_by_2.loc[distancias.loc[:,pedidos['ID_CLIENTE']
    [id_p]].idxmin(), 'LONGITUD']

    #Destino
    distribucion.loc[i, 'DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p], 'TIENDA']
    distribucion.loc[i, 'LAT_DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p], 'LATITUD']
    distribucion.loc[i, 'LONG_DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p], 'LONGITUD']

    #asignacion del id de la tienda
    id_tien = pedidos['ID_TIENDA'][id_p] #id de la tienda a partir de su nombre en matriz
    st_0_t = st_0[st_0['ID_TIENDA'] == id_tien] #Separa el stock por tienda
    ind = int(st_0_t.index.values[st_0_t['ID_ARTICULO'] == pedidos['ID_ARTICULO'][id_p]])

    if st_0.loc[ind, 'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
        distribucion.loc[i, 'DISPONIBILIDAD'] = 1
        distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
        distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
        v_cant = st_0.loc[ind, 'CANTIDAD'] - 1
        st_0.loc.__setitem__((ind, 'CANTIDAD'), v_cant)
        v_pre = st_0.loc[ind, 'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
        st_0.loc.__setitem__((ind, 'CANTIDAD_MONTO'), v_pre)

    else:
        distribucion.loc[i, 'DISPONIBILIDAD'] = 0
        distribucion.loc[i, 'CANTIDAD'] = 0
        distribucion.loc[i, 'PRECIO'] = 0
        coste_rot_t += pedidos['PRECIO'][id_p]
        n_rot_stock_t += 1

#Costes
coste_prep_t += c_p_t
else:
    #Origen
    for id_inv in range(min(st_0[st_0['ID_TIENDA'] == 15].index), max(st_0[st_0['ID_TIENDA'] == 15]
    .index)):

        if st_0['ID_ARTICULO'][id_inv] == pedidos['ID_ARTICULO'][id_p]:
            if st_0['CANTIDAD'][id_inv] > 0:
                distribucion.loc[i, 'DISPONIBILIDAD'] = 1
                distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
                distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
                v_cant = st_0.loc[id_inv, 'CANTIDAD'] - 1
                st_0.loc.__setitem__((id_inv, 'CANTIDAD'), v_cant)
                v_pre = st_0.loc[id_inv, 'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
                st_0.loc.__setitem__((id_inv, 'CANTIDAD_MONTO'), v_pre)
                distribucion.loc[i, 'ORIGEN'] = 'Almacen'
                distribucion.loc[i, 'LAT_ORIGEN'] = centros_by_1.loc['Almacen', 'LATITUD']
                distribucion.loc[i, 'LONG_ORIGEN'] = centros_by_1.loc['Almacen', 'LONGITUD']
                coste_transp_a += c_t * distancias.loc['Almacen', pedidos['ID_CLIENTE']
                [id_p]] #coste de transporte
                #Selección del coste de envío en función de la dimensión del paquete
                if articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p], 'DIM_PAQUETE'] == 'P':
                    c_env = c_ep
                elif articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p], 'DIM_PAQUETE'] == 'M':
                    c_env = c_em
                else:
                    c_env = c_eg

                coste_env_a += c_env

            else:
                distribucion.loc[i, 'DISPONIBILIDAD'] = 0
                distribucion.loc[i, 'CANTIDAD'] = 0
                distribucion.loc[i, 'PRECIO'] = 0
                coste_rot_a += pedidos['PRECIO'][id_p]
                n_rot_stock_a += 1

    #Origen
    distribucion.loc[i, 'ORIGEN'] = 'Almacen'
    distribucion.loc[i, 'LAT_ORIGEN'] = centros_by_1.loc['Almacen', 'LATITUD']
    distribucion.loc[i, 'LONG_ORIGEN'] = centros_by_1.loc['Almacen', 'LONGITUD']

```

```

#Destino
distribucion.loc[i,'DESTINO'] = "Domicilio cliente"
distribucion.loc[i,'LAT_DESTINO'] = clientes.loc[pedidos['ID_TIENDA'][id_p],'LATITUD']
distribucion.loc[i,'LONG_DESTINO'] = clientes.loc[pedidos['ID_TIENDA'][id_p],'LONGITUD']

#Costes
coste_prep_a += c_p_a

coste_total = coste_env_a + coste_env_t + coste_prep_t + coste_prep_a + coste_transp_t
              + coste_transp_a + coste_rot_t + coste_rot_a

i += 1
#print(distribucion)
print('Escenario 1: Distribuyendo pedido: ' + str(pedidos['ID_PEDIDO'][id_p]))

outputs['Valor'] = [int(coste_prep_t), int(coste_prep_a), int(coste_env_t), int(coste_env_a)
, int(coste_transp_t), int(coste_transp_a), int(coste_rot_t), int(coste_rot_a), int(coste_total)
, n_rot_stock_t, n_rot_stock_a]
outputs['Unidades'] = ['Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros'
, 'Uds', 'Uds']
outputs.index = ['Coste Mano de Obra Tienda', 'Coste Mano de Obra Almacen', 'Coste Packaging Tienda'
, 'Coste Packaging Almacen', 'Coste Transporte Tienda', 'Coste Transporte Almacen'
, 'Coste Rotura Stock Tienda', 'Coste Rotura Stock Almacen', 'Coste Total'
, 'Pedidos no satisfechos tienda', 'Pedidos no satisfechos almacen']

if inputs['INPUTS']['Caso_1'] == 'Si':
    outputs.to_excel('Outputs_E1_C1.xlsx')
    distribucion.to_excel('OMS_E1_C1.xlsx', index = False)
if inputs['INPUTS']['Caso_2'] == 'Si':
    outputs.to_excel('Outputs_E1_C2.xlsx')
    distribucion.to_excel('OMS_E1_C2.xlsx', index = False)

```


11.5. Anexo 5: OMS. Escenario 2.

#OMS. Escenario 2

```
import pandas as pd
import numpy as np

#Cargar maestros y BBDDs
pedidos = pd.read_excel('Pedidos.xlsx')
clientes = pd.read_excel('Clientes.xlsx', index_col = 0)
inputs = pd.read_excel('Inputs.xlsx', index_col = 0)

if inputs['INPUTS']['Caso_1'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C1.xlsx')
    st_0 = pd.read_excel('Stock_E2_C1.xlsx')
if inputs['INPUTS']['Caso_2'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C2.xlsx')
    st_0 = pd.read_excel('Stock_E2_C2.xlsx')

centros = pd.read_excel('Centros.xlsx')
articulos = pd.read_excel('Articulos.xlsx')
devoluciones = pd.read_excel('Devoluciones.xlsx')
distancias = pd.read_excel('Distancias.xlsx', index_col = 0)
costes = pd.read_excel('Costes.xlsx', index_col = 0)

#Inputs
#Stock seguridad
st_seg = tiendas['STOCK_SEGURIDAD']

#Costes
c_p_t = costes['COSTE']['Coste_preparacion_t']
c_p_a = costes['COSTE']['Coste_preparacion_a']
c_t = costes['COSTE']['Coste_transporte']
c_ep = costes['COSTE']['Coste_envio_p']
c_em = costes['COSTE']['Coste_envio_m']
c_eg = costes['COSTE']['Coste_envio_g']

#Definicion de dataframes
cols_d = ['ID_PEDIDO', 'FECHA_PEDIDO', 'DISPONIBILIDAD', 'ORIGEN', 'LAT_ORIGEN', 'LONG_ORIGEN',
          'DESTINO', 'LAT_DESTINO', 'LONG_DESTINO', 'ID_CLIENTE', 'ID_ARTICULO', 'CANTIDAD', 'PRECIO']
distribucion = pd.DataFrame(columns = cols_d) #DF de asignacion de origenes de pedidos
outputs = pd.DataFrame(columns = ['Valor', 'Unidades']) #DF de outputs

#Definicion de outputs
cont = 0
coste_prep_t = 0
coste_prep_a = 0
coste_transp_t = 0
coste_transp_a = 0
coste_env_t = 0
coste_env_a = 0
coste_rot_t = 0
coste_rot_a = 0
coste_total = 0
n_rot_stock_t = 0
n_rot_stock_a = 0

#Filtrar dataframes
tiendas_by_1 = tiendas.set_index('ID_TIENDA')
tiendas_by_2 = tiendas.set_index('TIENDA')
centros_by_1 = centros.set_index('CENTROS')
articulos_by_1 = articulos.set_index('ID_ARTICULO')

i = 0 #Contador para el DF
for id_p in range(0, len(pedidos)):
    distribucion.loc[i, 'ID_PEDIDO'] = pedidos['ID_PEDIDO'][id_p]
    distribucion.loc[i, 'FECHA_PEDIDO'] = pedidos['FECHA_PEDIDO'][id_p]
    distribucion.loc[i, 'ID_CLIENTE'] = pedidos['ID_CLIENTE'][id_p]
    distribucion.loc[i, 'ID_ARTICULO'] = pedidos['ID_ARTICULO'][id_p]

    dist_min = distancias[pedidos['ID_CLIENTE'][id_p]].sort_values(ascending = True)
    #Ordena las tiendas por cercania segun el pedido

    #Seleccion de los datos de origen y destino en funcion de la tienda.
    if pedidos['ID_TIENDA'][id_p] != 15:
```

```

#Origen
distribucion.loc[i, 'ORIGEN'] = distancias.loc[:,pedidos['ID_CLIENTE'][id_p]].idxmin()
distribucion.loc[i, 'LAT_ORIGEN'] = tiendas_by_2.loc[distancias.loc[:,pedidos['ID_CLIENTE']
[id_p]].idxmin(),'LATITUD']
distribucion.loc[i, 'LONG_ORIGEN'] = tiendas_by_2.loc[distancias.loc[:,pedidos['ID_CLIENTE']
[id_p]].idxmin(),'LONGITUD']
#Destino
distribucion.loc[i, 'DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p],'TIENDA']
distribucion.loc[i, 'LAT_DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p],'LATITUD']
distribucion.loc[i, 'LONG_DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p],'LONGITUD']

#asignacion del id de la tienda
id_tien = pedidos['ID_TIENDA'][id_p] #id de la tienda a partir de su nombre en matriz
st_0_t = st_0[st_0['ID_TIENDA'] == id_tien] #Separa el stock por tienda
ind = int(st_0_t.index.values[st_0_t['ID_ARTICULO'] == pedidos['ID_ARTICULO'][id_p]])

if st_0.loc[ind,'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
    distribucion.loc[i, 'DISPONIBILIDAD'] = 1
    distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
    distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
    v_cant = st_0.loc[ind,'CANTIDAD'] - 1
    st_0.loc.__setitem__((ind,'CANTIDAD'), v_cant)
    v_pre = st_0.loc[ind,'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
    st_0.loc.__setitem__((ind,'CANTIDAD'), v_pre)

else:
    distribucion.loc[i, 'DISPONIBILIDAD'] = 0
    distribucion.loc[i, 'CANTIDAD'] = 0
    distribucion.loc[i, 'PRECIO'] = 0
    coste_rot_t += pedidos['PRECIO'][id_p]
    n_rot_stock_t += 1

#Costes
coste_prep_t += c_p_t
else:
    respuesta = True

#Origen
for j in range(0, len(dist_min)-1):
    #localiza la tienda/centro mas cercana que tiene stock suficiente
    rot = 0 #Variable que caracteriza la rotura

    while respuesta == True:
        if dist_min.index[j] != 'Almacen':
            id_t = tiendas_by_2.loc[dist_min.index[j], 'ID_TIENDA']
            #id de la tienda a partir de su nombre en matriz
            st_0_t = st_0[st_0['ID_TIENDA'] == id_t] #Separa el stock por tienda
            ind = int(st_0_t.index.values[st_0_t['ID_ARTICULO'] ==
            pedidos['ID_ARTICULO'][id_p]])

            if st_0_t.loc[ind,'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
                distribucion.loc[i, 'ORIGEN'] = dist_min.index[j]
                distribucion.loc[i, 'LAT_ORIGEN'] = tiendas_by_1.loc[id_t
                , 'LATITUD']
                distribucion.loc[i, 'LONG_ORIGEN'] = tiendas_by_1.loc[id_t
                , 'LONGITUD']
                distribucion.loc[i, 'DISPONIBILIDAD'] = 1
                distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
                distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
                v_cant = st_0_t.loc[ind,'CANTIDAD'] - 1
                st_0_t.loc.__setitem__((ind,'CANTIDAD'), v_cant)
                v_pre = st_0_t.loc[ind,'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
                st_0_t.loc.__setitem__((ind,'CANTIDAD'), v_pre)
                coste_transp_a += c_t * distancias.loc[dist_min.index[j]
                ,pedidos['ID_CLIENTE'][id_p]]
                #coste de transporte
                coste_prep_t += c_p_t
                respuesta = False
            else:
                j += 1

        elif dist_min.index[j] == 'Almacen':
            st_0_a = st_0[st_0['ID_TIENDA'] == 15] #Separa el stock por tienda

```

```

ind = int(st_0_a.index.values[st_0_a['ID_ARTICULO'] ==
pedidos['ID_ARTICULO'][id_p]])
if st_0_a.loc[ind, 'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
    distribucion.loc[i, 'ORIGEN'] = 'Almacen'
    distribucion.loc[i, 'LAT_ORIGEN'] = centros_by_1.loc['Almacen'
, 'LATITUD']
    distribucion.loc[i, 'LONG_ORIGEN'] = centros_by_1.loc['Almacen'
, 'LONGITUD']
    distribucion.loc[i, 'DISPONIBILIDAD'] = 1
    distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
    distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
    v_cant = st_0.loc[ind, 'CANTIDAD'] - 1
    st_0.loc.__setitem__((ind, 'CANTIDAD'), v_cant)
    v_pre = st_0.loc[ind, 'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
    st_0.loc.__setitem__((ind, 'CANTIDAD'), v_pre)
    coste_transp_a += c_t * distancias.loc['Almacen'
, pedidos['ID_CLIENTE'][id_p]] #coste de transporte
    coste_prep_a += c_p_a
    respuesta = False
else:
    j += 1

else:
    distribucion.loc[i, 'DISPONIBILIDAD'] = 0
    distribucion.loc[i, 'CANTIDAD'] = 0
    distribucion.loc[i, 'PRECIO'] = 0
    coste_rot_a += pedidos['PRECIO'][id_p]
    n_rot_stock_a += 1
    respuesta = False

#Destino
distribucion.loc[i, 'DESTINO'] = "Domicilio cliente"
distribucion.loc[i, 'LAT_DESTINO'] = clientes.loc[pedidos['ID_TIENDA']
[id_p], 'LATITUD']
distribucion.loc[i, 'LONG_DESTINO'] = clientes.loc[pedidos['ID_TIENDA']
[id_p], 'LONGITUD']

#Selección del coste de envío en función de la dimensión del paquete
if articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p], 'DIM_PAQUETE'] == 'P':
    c_env = c_ep
elif articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p], 'DIM_PAQUETE'] == 'M':
    c_env = c_em
else:
    c_env = c_eg

coste_env_a += c_env

coste_total = coste_env_a + coste_env_t + coste_prep_t + coste_prep_a
+ coste_transp_t + coste_transp_a + coste_rot_t + coste_rot_a

i += 1
#print(distribucion)
print('Escenario 2: Distribuyendo pedido: ' + str(pedidos['ID_PEDIDO'][id_p]))

outputs['Valor'] = [int(coste_prep_t), int(coste_prep_a), int(coste_env_t), int(coste_env_a)
, int(coste_transp_t), int(coste_transp_a), int(coste_rot_t), int(coste_rot_a), int(coste_total)
, n_rot_stock_t, n_rot_stock_a]
outputs['Unidades'] = ['Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros'
, 'Euros', 'Uds', 'Uds']
outputs.index = ['Coste Mano de Obra Tienda', 'Coste Mano de Obra Almacen', 'Coste Packaging Tienda'
, 'Coste Packaging Almacen', 'Coste Transporte Tienda', 'Coste Transporte Almacen'
, 'Coste Rotura Stock Tienda', 'Coste Rotura Stock Almacen', 'Coste Total'
, 'Pedidos no satisfechos tienda', 'Pedidos no satisfechos almacen']

if inputs['INPUTS']['Caso_1'] == 'Si':
    outputs.to_excel('Outputs_E2_C1.xlsx')
    distribucion.to_excel('OMS_E2_C1.xlsx', index = False)
if inputs['INPUTS']['Caso_2'] == 'Si':
    outputs.to_excel('Outputs_E2_C2.xlsx')
    distribucion.to_excel('OMS_E2_C2.xlsx', index = False)

```


11.6. Anexo 6: OMS. Escenario 3.

#OMS. Escenario 3

```
import pandas as pd
import numpy as np

#Cargar maestros y BBDDs
pedidos = pd.read_excel('Pedidos.xlsx')
clientes = pd.read_excel('Clientes.xlsx', index_col = 0)
inputs = pd.read_excel('Inputs.xlsx', index_col = 0)

if inputs['INPUTS']['Caso_1'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C1.xlsx')
    st_0 = pd.read_excel('Stock_E2_C1.xlsx')
if inputs['INPUTS']['Caso_2'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C2.xlsx')
    st_0 = pd.read_excel('Stock_E2_C2.xlsx')

centros = pd.read_excel('Centros.xlsx')
articulos = pd.read_excel('Articulos.xlsx')
devoluciones = pd.read_excel('Devoluciones.xlsx')
distancias = pd.read_excel('Distancias.xlsx', index_col = 0)
costes = pd.read_excel('Costes.xlsx', index_col = 0)

#Inputs
#Stock seguridad
st_seg = tiendas['STOCK_SEGURIDAD']

#Costes
c_p_t = costes['COSTE']['Coste_preparacion_t']
c_p_a = costes['COSTE']['Coste_preparacion_a']
c_t = costes['COSTE']['Coste_transporte']
c_ep = costes['COSTE']['Coste_envio_p']
c_em = costes['COSTE']['Coste_envio_m']
c_eg = costes['COSTE']['Coste_envio_g']

#Definicion de dataframes
cols_d = ['ID_PEDIDO', 'FECHA_PEDIDO', 'DISPONIBILIDAD', 'ORIGEN', 'LAT_ORIGEN', 'LONG_ORIGEN',
          'DESTINO', 'LAT_DESTINO', 'LONG_DESTINO', 'ID_CLIENTE', 'ID_ARTICULO', 'CANTIDAD', 'PRECIO']
distribucion = pd.DataFrame(columns = cols_d) #DF de asignacion de origenes de pedidos
outputs = pd.DataFrame(columns = ['Valor', 'Unidades', 'Porcentaje']) #DF de outputs

#Definicion de outputs
cont = 0
coste_prep_t = 0
coste_prep_a = 0
coste_transp_t = 0
coste_transp_a = 0
coste_env_t = 0
coste_env_a = 0
coste_rot_t = 0
coste_rot_a = 0
coste_total = 0
n_rot_stock_t = 0
n_rot_stock_a = 0

#Filtrar dataframes
tiendas_by_1 = tiendas.set_index('ID_TIENDA')
tiendas_by_2 = tiendas.set_index('TIENDA')
centros_by_1 = centros.set_index('CENTROS')
articulos_by_1 = articulos.set_index('ID_ARTICULO')

i = 0 #Contador para el DF
for id_p in range(0, len(pedidos)):
    distribucion.loc[i, 'ID_PEDIDO'] = pedidos['ID_PEDIDO'][id_p]
    distribucion.loc[i, 'FECHA_PEDIDO'] = pedidos['FECHA_PEDIDO'][id_p]
    distribucion.loc[i, 'ID_CLIENTE'] = pedidos['ID_CLIENTE'][id_p]
    distribucion.loc[i, 'ID_ARTICULO'] = pedidos['ID_ARTICULO'][id_p]

    dist_min = distancias[pedidos['ID_CLIENTE'][id_p]].sort_values(ascending = True)
    #Ordena las tiendas por cercania segun el pedido

    #Seleccion de los datos de origen y destino en funcion de la tienda.
    if pedidos['ID_TIENDA'][id_p] != 15:
```

```

#Origen
distribucion.loc[i, 'ORIGEN'] = distancias.loc[:,pedidos['ID_CLIENTE'][id_p]].idxmin()
distribucion.loc[i, 'LAT_ORIGEN'] = tiendas_by_2.loc[distancias.loc[:,pedidos['ID_CLIENTE']
[id_p]].idxmin(),'LATITUD']
distribucion.loc[i, 'LONG_ORIGEN'] = tiendas_by_2.loc[distancias.loc[:,pedidos['ID_CLIENTE']
[id_p]].idxmin(),'LONGITUD']
#Destino
distribucion.loc[i, 'DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p],'TIENDA']
distribucion.loc[i, 'LAT_DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p],'LATITUD']
distribucion.loc[i, 'LONG_DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p],'LONGITUD']

#asignacion del id de la tienda
id_tien = pedidos['ID_TIENDA'][id_p] #id de la tienda a partir de su nombre en matriz
st_0_t = st_0[st_0['ID_TIENDA'] == id_tien] #Separa el stock por tienda
ind = int(st_0_t.index.values[st_0_t['ID_ARTICULO'] == pedidos['ID_ARTICULO'][id_p]])

if st_0.loc[ind,'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
    distribucion.loc[i, 'DISPONIBILIDAD'] = 1
    distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
    distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
    v_cant = st_0.loc[ind,'CANTIDAD'] - 1
    st_0.loc.__setitem__((ind,'CANTIDAD'), v_cant)
    v_pre = st_0.loc[ind,'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
    st_0.loc.__setitem__((ind,'CANTIDAD'), v_pre)

else:
    distribucion.loc[i, 'DISPONIBILIDAD'] = 0
    distribucion.loc[i, 'CANTIDAD'] = 0
    distribucion.loc[i, 'PRECIO'] = 0
    coste_rot_t += pedidos['PRECIO'][id_p]
    n_rot_stock_t += 1

#Costes
coste_prep_t += c_p_t
else:
    respuesta = True

#Origen
for j in range(0, len(dist_min)-1):
    #localiza la tienda/centro mas cercana que tiene stock suficiente
    rot = 0 #Variable que caracteriza la rotura

    while respuesta == True:
        if dist_min.index[j] != 'Almacen':
            id_t = tiendas_by_2.loc[dist_min.index[j], 'ID_TIENDA']
            #id de la tienda a partir de su nombre en matriz
            st_0_t = st_0[st_0['ID_TIENDA'] == id_t] #Separa el stock por tienda
            ind = int(st_0_t.index.values[st_0_t['ID_ARTICULO'] ==
pedidos['ID_ARTICULO'][id_p]])

            if st_0_t.loc[ind,'CANTIDAD'] > st_seg[id_t - 1]: #Comprueba que hay stock
                distribucion.loc[i, 'ORIGEN'] = dist_min.index[j]
                distribucion.loc[i, 'LAT_ORIGEN'] = tiendas_by_1.loc[id_t
, 'LATITUD']
                distribucion.loc[i, 'LONG_ORIGEN'] = tiendas_by_1.loc[id_t
, 'LONGITUD']
                distribucion.loc[i, 'DISPONIBILIDAD'] = 1
                distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
                distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
                v_cant = st_0.loc[ind,'CANTIDAD'] - 1
                st_0.loc.__setitem__((ind,'CANTIDAD'), v_cant)
                v_pre = st_0.loc[ind,'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
                st_0.loc.__setitem__((ind,'CANTIDAD'), v_pre)
                coste_transp_a += c_t * distancias.loc[dist_min.index[j]
,pedidos['ID_CLIENTE'][id_p]] #coste de transporte
                coste_prep_t += c_p_t
                respuesta = False
            else:
                j += 1

        elif dist_min.index[j] == 'Almacen':
            st_0_a = st_0[st_0['ID_TIENDA'] == 15] #Separa el stock por tienda
            ind = int(st_0_a.index.values[st_0_a['ID_ARTICULO'] ==

```

```

pedidos['ID_ARTICULO'][id_p])
if st_0_a.loc[ind,'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
    distribucion.loc[i, 'ORIGEN'] = 'Almacen'
    distribucion.loc[i, 'LAT_ORIGEN'] = centros_by_1.loc['Almacen', 'LATITUD']
    distribucion.loc[i, 'LONG_ORIGEN'] = centros_by_1.loc['Almacen', 'LONGITUD']
    distribucion.loc[i, 'DISPONIBILIDAD'] = 1
    distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
    distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
    v_cant = st_0.loc[ind, 'CANTIDAD'] - 1
    st_0.loc.__setitem__((ind, 'CANTIDAD'), v_cant)
    v_pre = st_0.loc[ind, 'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
    st_0.loc.__setitem__((ind, 'CANTIDAD'), v_pre)
    coste_transp_a += c_t * distancias.loc['Almacen', pedidos['ID_CLIENTE'][id_p]] #coste de transporte
    coste_prep_a += c_p_a
    respuesta = False
else:
    j += 1

else:
    distribucion.loc[i, 'DISPONIBILIDAD'] = 0
    distribucion.loc[i, 'CANTIDAD'] = 0
    distribucion.loc[i, 'PRECIO'] = 0
    coste_rot_a += pedidos['PRECIO'][id_p]
    n_rot_stock_a += 1
    respuesta = False

#Destino
distribucion.loc[i, 'DESTINO'] = "Domicilio cliente"
distribucion.loc[i, 'LAT_DESTINO'] = clientes.loc[pedidos['ID_TIENDA'][id_p], 'LATITUD']
distribucion.loc[i, 'LONG_DESTINO'] = clientes.loc[pedidos['ID_TIENDA'][id_p], 'LONGITUD']

#Selección del coste de envío en función de la dimensión del paquete
if articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p], 'DIM_PAQUETE'] == 'P':
    c_env = c_ep
elif articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p], 'DIM_PAQUETE'] == 'M':
    c_env = c_em
else:
    c_env = c_eg

coste_env_a += c_env

coste_total = coste_env_a + coste_env_t + coste_prep_t + coste_prep_a
+ coste_transp_t + coste_transp_a + coste_rot_t + coste_rot_a

i += 1
#print(distribucion)
print('Escenario 3: Distribuyendo pedido: ' + str(pedidos['ID_PEDIDO'][id_p]))

outputs['Valor'] = [int(coste_prep_t), int(coste_prep_a), int(coste_env_t), int(coste_env_a),
int(coste_transp_t), int(coste_transp_a), int(coste_rot_t), int(coste_rot_a), int(coste_total),
n_rot_stock_t, n_rot_stock_a]
outputs['Unidades'] = ['Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros',
'Euros', 'Euros', 'Uds', 'Uds']
outputs.index = ['Coste Mano de Obra Tienda', 'Coste Mano de Obra Almacen', 'Coste Packaging Tienda',
'Coste Packaging Almacen', 'Coste Transporte Tienda', 'Coste Transporte Almacen',
'Coste Rotura Stock Tienda', 'Coste Rotura Stock Almacen', 'Coste Total',
'Pedidos no satisfechos tienda', 'Pedidos no satisfechos almacen']

if inputs['INPUTS']['Caso_1'] == 'Si':
    outputs.to_excel('Outputs_E3_C1.xlsx')
    distribucion.to_excel('OMS_E3_C1.xlsx', index = False)
if inputs['INPUTS']['Caso_2'] == 'Si':
    outputs.to_excel('Outputs_E3_C2.xlsx')
    distribucion.to_excel('OMS_E3_C2.xlsx', index = False)

```


11.7. Anexo 7: OMS. Escenario 4.

#OMS. Escenario 4

```
import pandas as pd
import numpy as np

#Cargar maestros y BBDDs
pedidos = pd.read_excel('Pedidos.xlsx')
clientes = pd.read_excel('Clientes.xlsx', index_col = 0)
inputs = pd.read_excel('Inputs.xlsx', index_col = 0)

if inputs['INPUTS']['Caso_1'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C1.xlsx')
    st_0 = pd.read_excel('Stock_E2_C1.xlsx')
if inputs['INPUTS']['Caso_2'] == 'Si':
    tiendas = pd.read_excel('Tiendas_C2.xlsx')
    st_0 = pd.read_excel('Stock_E2_C2.xlsx')

centros = pd.read_excel('Centros.xlsx')
articulos = pd.read_excel('Articulos.xlsx')
devoluciones = pd.read_excel('Devoluciones.xlsx')
distancias = pd.read_excel('Distancias.xlsx', index_col = 0)
costes = pd.read_excel('Costes.xlsx', index_col = 0)

#Inputs
#Stock seguridad
st_seg = tiendas['STOCK_SEGURIDAD']

#Costes
c_p_t = costes['COSTE']['Coste_preparacion_t']
c_p_a = costes['COSTE']['Coste_preparacion_a']
c_t = costes['COSTE']['Coste_transporte']
c_ep = costes['COSTE']['Coste_envio_p']
c_em = costes['COSTE']['Coste_envio_m']
c_eg = costes['COSTE']['Coste_envio_g']

#Definicion de dataframes
cols_d = ['ID_PEDIDO', 'FECHA_PEDIDO', 'DISPONIBILIDAD', 'ORIGEN', 'LAT_ORIGEN', 'LONG_ORIGEN',
          'DESTINO', 'LAT_DESTINO', 'LONG_DESTINO', 'ID_CLIENTE', 'ID_ARTICULO', 'CANTIDAD', 'PRECIO']
distribucion = pd.DataFrame(columns = cols_d) #DF de asignacion de origenes de pedidos
outputs = pd.DataFrame(columns = ['Valor', 'Unidades', 'Porcentaje']) #DF de outputs

#Definicion de outputs
cont = 0
coste_prep_t = 0
coste_prep_a = 0
coste_transp_t = 0
coste_transp_a = 0
coste_env_t = 0
coste_env_a = 0
coste_rot_t = 0
coste_rot_a = 0
coste_total = 0
n_rot_stock_t = 0
n_rot_stock_a = 0

#Filtrar dataframes
tiendas_by_1 = tiendas.set_index('ID_TIENDA')
tiendas_by_2 = tiendas.set_index('TIENDA')
centros_by_1 = centros.set_index('CENTROS')
articulos_by_1 = articulos.set_index('ID_ARTICULO')

i = 0 #Contador para el DF
for id_p in range(0, len(pedidos)):
    distribucion.loc[i, 'ID_PEDIDO'] = pedidos['ID_PEDIDO'][id_p]
    distribucion.loc[i, 'FECHA_PEDIDO'] = pedidos['FECHA_PEDIDO'][id_p]
    distribucion.loc[i, 'ID_CLIENTE'] = pedidos['ID_CLIENTE'][id_p]
    distribucion.loc[i, 'ID_ARTICULO'] = pedidos['ID_ARTICULO'][id_p]

    dist_min = distancias[pedidos['ID_CLIENTE'][id_p]].sort_values(ascending = True)
    #Ordena las tiendas por cercania segun el pedido

    #Seleccion de los datos de origen y destino en funcion de la tienda.
    if pedidos['ID_TIENDA'][id_p] != 15:
        respuesta = True
```

```

inter_t = np.random.randint(0,2)
for j in range(0, len(dist_min)-1):
    #localiza la tienda/centro mas cercana que tiene stock suficiente
    while respuesta == True:
        if dist_min.index[j] != 'Almacen':
            id_t = tiendas_by_2.loc[dist_min.index[j], 'ID_TIENDA']
            #id de la tienda a partir de su nombre en matriz
            st_0_t = st_0[st_0['ID_TIENDA'] == id_t] #Separa el stock por tienda
            ind = int(st_0_t.index.values[st_0_t['ID_ARTICULO'] ==
            pedidos['ID_ARTICULO'][id_p]])
            if st_0_t.loc[ind, 'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
                distribucion.loc[i, 'ORIGEN'] = dist_min.index[j]
                distribucion.loc[i, 'LAT_ORIGEN'] = tiendas_by_1.loc[id_t,
                'LATITUD']
                distribucion.loc[i, 'LONG_ORIGEN'] = tiendas_by_1.loc[id_t,
                'LONGITUD']
                distribucion.loc[i, 'DISPONIBILIDAD'] = 1
                distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
                distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
                v_cant = st_0.loc[ind, 'CANTIDAD'] - 1
                st_0.loc.__setitem__((ind, 'CANTIDAD'), v_cant)
                v_pre = st_0.loc[ind, 'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
                st_0.loc.__setitem__((ind, 'CANTIDAD_MONTO'), v_pre)
                coste_transp_t += c_t * distancias.loc[dist_min.index[j],
                pedidos['ID_CLIENTE'][id_p]] #coste de transporte
                respuesta = False
                if dist_min.index[j] != tiendas_by_1.loc[pedidos['ID_TIENDA']
                [id_p], 'TIENDA']:
                    #Selección del coste de envío en
                    #función de la dimensión del paquete
                    if articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p],
                    'DIM_PAQUETE'] == 'P':
                        c_env = c_ep
                    elif articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p],
                    'DIM_PAQUETE'] == 'M':
                        c_env = c_em
                    else:
                        c_env = c_eg
                    coste_env_t += c_env
                coste_prep_t += c_p_t
            else:
                if inter_t == 1:
                    distribucion.loc[i, 'DISPONIBILIDAD'] = 0
                    distribucion.loc[i, 'CANTIDAD'] = 0
                    distribucion.loc[i, 'PRECIO'] = 0
                    coste_rot_t += pedidos['PRECIO'][id_p]
                    n_rot_stock_t += 1
                    respuesta = False
                else:
                    j += 1
        elif dist_min.index[j] == 'Almacen':
            st_0_a = st_0[st_0['ID_TIENDA'] == 15] #Separa el stock por tienda
            ind = int(st_0_a.index.values[st_0_a['ID_ARTICULO'] ==
            pedidos['ID_ARTICULO'][id_p]])
            if st_0_a.loc[ind, 'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
                distribucion.loc[i, 'ORIGEN'] = 'Almacen'
                distribucion.loc[i, 'LAT_ORIGEN'] = centros_by_1.loc['Almacen',
                'LATITUD']
                distribucion.loc[i, 'LONG_ORIGEN'] = centros_by_1.loc['Almacen',
                'LONGITUD']
                distribucion.loc[i, 'DISPONIBILIDAD'] = 1
                distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
                distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
                v_cant = st_0.loc[ind, 'CANTIDAD'] - 1
                st_0.loc.__setitem__((ind, 'CANTIDAD'), v_cant)
                v_pre = st_0.loc[ind, 'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
                st_0.loc.__setitem__((ind, 'CANTIDAD_MONTO'), v_pre)
                coste_transp_t += c_t * distancias.loc['Almacen',
                pedidos['ID_CLIENTE'][id_p]] #coste de transporte
                respuesta = False
                #Selección del coste de envío en
                #función de la dimensión del paquete

```

```

        if articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p],
'DIM_PAQUETE'] == 'P':
            c_env = c_ep
        elif articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p],
'DIM_PAQUETE'] == 'M':
            c_env = c_em
        else:
            c_env = c_eg

        coste_env_t += c_env
        coste_prep_a += c_p_a
    else:
        if inter_t == 1:
            distribucion.loc[i, 'DISPONIBILIDAD'] = 0
            distribucion.loc[i, 'CANTIDAD'] = 0
            distribucion.loc[i, 'PRECIO'] = 0
            coste_rot += pedidos['PRECIO'][id_p]
            n_rot_stock_t += 1
            respuesta = False
        else:
            j += 1

    else:
        distribucion.loc[i, 'DISPONIBILIDAD'] = 0
        distribucion.loc[i, 'CANTIDAD'] = 0
        distribucion.loc[i, 'PRECIO'] = 0
        coste_rot_t += pedidos['PRECIO'][id_p]
        n_rot_stock_t += 1
        respuesta = False

#Destino
distribucion.loc[i, 'DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p], 'TIENDA']
distribucion.loc[i, 'LAT_DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p], 'LATITUD']
distribucion.loc[i, 'LONG_DESTINO'] = tiendas_by_1.loc[pedidos['ID_TIENDA'][id_p], 'LONGITUD']

else:
    respuesta = True

#Origen
for j in range(0, len(dist_min)-1):
    #localiza la tienda/centro mas cercana que tiene stock suficiente
    rot = 0 #Variable que caracteriza la rotura

    while respuesta == True:
        if dist_min.index[j] != 'Almacen':
            id_t = tiendas_by_2.loc[dist_min.index[j], 'ID_TIENDA']
            #id de la tienda a partir de su nombre en matriz
            st_0_t = st_0[st_0['ID_TIENDA'] == id_t] #Separa el stock por tienda
            ind = int(st_0_t.index.values[st_0_t['ID_ARTICULO'] ==
pedidos['ID_ARTICULO'][id_p]])

            if st_0_t.loc[ind, 'CANTIDAD'] > st_seg[id_t - 1]:
                #Comprueba que hay stock suficiente
                distribucion.loc[i, 'ORIGEN'] = dist_min.index[j]
                distribucion.loc[i, 'LAT_ORIGEN'] = tiendas_by_1.loc[id_t,
'LATITUD']
                distribucion.loc[i, 'LONG_ORIGEN'] = tiendas_by_1.loc[id_t,
'LONGITUD']
                distribucion.loc[i, 'DISPONIBILIDAD'] = 1
                distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
                distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
                v_cant = st_0.loc[ind, 'CANTIDAD'] - 1
                st_0.loc.__setitem__((ind, 'CANTIDAD'), v_cant)
                v_pre = st_0.loc[ind, 'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
                st_0.loc.__setitem__((ind, 'CANTIDAD'), v_pre)
                coste_transp_a += c_t * distancias.loc[dist_min.index[j],
pedidos['ID_CLIENTE'][id_p]] #coste de transporte
                respuesta = False
                coste_prep_t += c_p_t
            else:
                j += 1

        elif dist_min.index[j] == 'Almacen':
            st_0_a = st_0[st_0['ID_TIENDA'] == 15] #Separa el stock por tienda

```

```

ind = int(st_0_a.index.values[st_0_a['ID_ARTICULO'] ==
pedidos['ID_ARTICULO'][id_p]])
if st_0_a.loc[ind, 'CANTIDAD'] > 0: #Comprueba que hay stock suficiente
    distribucion.loc[i, 'ORIGEN'] = 'Almacen'
    distribucion.loc[i, 'LAT_ORIGEN'] = centros_by_1.loc['Almacen'
, 'LATITUD']
    distribucion.loc[i, 'LONG_ORIGEN'] = centros_by_1.loc['Almacen'
, 'LONGITUD']
    distribucion.loc[i, 'DISPONIBILIDAD'] = 1
    distribucion.loc[i, 'CANTIDAD'] = pedidos['CANTIDAD'][id_p]
    distribucion.loc[i, 'PRECIO'] = pedidos['PRECIO'][id_p]
    v_cant = st_0.loc[ind, 'CANTIDAD'] - 1
    st_0.loc.__setitem__((ind, 'CANTIDAD'), v_cant)
    v_pre = st_0.loc[ind, 'CANTIDAD_MONTO'] - pedidos['PRECIO'][id_p]
    st_0.loc.__setitem__((ind, 'CANTIDAD'), v_pre)
    coste_transp_a += c_t * distancias.loc['Almacen'
, pedidos['ID_CLIENTE'][id_p]] #coste de transporte
    respuesta = False
    coste_prep_a += c_p_a
else:
    j += 1

else:
    distribucion.loc[i, 'DISPONIBILIDAD'] = 0
    distribucion.loc[i, 'CANTIDAD'] = 0
    distribucion.loc[i, 'PRECIO'] = 0
    coste_rot_a += pedidos['PRECIO'][id_p]
    n_rot_stock_a += 1
    respuesta = False

#Destino
distribucion.loc[i, 'DESTINO'] = "Domicilio cliente"
distribucion.loc[i, 'LAT_DESTINO'] = clientes.loc[pedidos['ID_TIENDA'][id_p], 'LATITUD']
distribucion.loc[i, 'LONG_DESTINO'] = clientes.loc[pedidos['ID_TIENDA'][id_p], 'LONGITUD']

#Selección del coste de envío en función de la dimensión del paquete
if articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p], 'DIM_PAQUETE'] == 'P':
    c_env = c_ep
elif articulos_by_1.loc[pedidos['ID_ARTICULO'][id_p], 'DIM_PAQUETE'] == 'M':
    c_env = c_em
else:
    c_env = c_eg

coste_env_a += c_env

coste_total = coste_env_a + coste_env_t + coste_prep_t + coste_prep_a
+ coste_transp_t + coste_transp_a + coste_rot_t + coste_rot_a

i += 1
#print(distribucion)
print('Escenario 4: Distribuyendo pedido: ' + str(pedidos['ID_PEDIDO'][id_p]))

outputs['Valor'] = [int(coste_prep_t), int(coste_prep_a), int(coste_env_t), int(coste_env_a)
, int(coste_transp_t), int(coste_transp_a), int(coste_rot_t), int(coste_rot_a)
, int(coste_total), n_rot_stock_t, n_rot_stock_a]
outputs['Unidades'] = ['Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros', 'Euros'
, 'Euros', 'Euros', 'Uds', 'Uds']
outputs.index = ['Coste Mano de Obra Tienda', 'Coste Mano de Obra Almacen', 'Coste Packaging Tienda'
, 'Coste Packaging Almacen', 'Coste Transporte Tienda', 'Coste Transporte Almacen'
, 'Coste Rotura Stock Tienda', 'Coste Rotura Stock Almacen', 'Coste Total'
, 'Pedidos no satisfechos tienda', 'Pedidos no satisfechos almacen']

if inputs['INPUTS']['Caso_1'] == 'Si':
    outputs.to_excel('Outputs_E4_C1.xlsx')
    distribucion.to_excel('OMS_E4_C1.xlsx', index = False)
if inputs['INPUTS']['Caso_2'] == 'Si':
    outputs.to_excel('Outputs_E4_C2.xlsx')
    distribucion.to_excel('OMS_E4_C2.xlsx', index = False)

```

11.8. Anexo 8: Generador histórico de pedidos

#Generador histórico de pedidos

```
import pandas as pd
import numpy as np
import random
from datetime import datetime, timedelta

##BBDD
articulos = pd.read_excel('Articulos.xlsx')
clientes = pd.read_excel('Clientes.xlsx')
tiendas = pd.read_excel('Tiendas.xlsx')
ventas_tiendas = pd.read_excel('Ventas.xlsx')
precios_art = pd.read_excel('Historico_precios.xlsx')
porcen_art = pd.read_excel('Distribucion_art.xlsx')

#Inicializacion
cont = 0
contador = 1
data = {'ID_PEDIDO': [], 'FECHA_PEDIDO': [], 'ID_TIENDA': [], 'ID_CLIENTE': [], ...
... 'ID_ARTICULO': [], 'CANTIDAD': [], 'COSTE': [], 'MARGEN': [], 'PRECIO': []}
cols = ['ID_PEDIDO', 'FECHA_PEDIDO', 'ID_TIENDA', 'ID_CLIENTE', 'ID_ARTICULO', ...
... 'CANTIDAD', 'COSTE', 'MARGEN', 'PRECIO']
art_gen = []

ventas = ventas_tiendas[['Tienda 1', 'Tienda 2', 'Tienda 3', 'Tienda 4']]
p_art = porcen_art[[1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009]]
precios = precios_art[[1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009]]

pedidos = pd.DataFrame(columns = cols)

np.random.seed(0)

for dia in range(0, len(ventas_tiendas)):

    #Ventas diarias
    aux_v = [0, 0, 0, 0]
    aux_v1 = ventas[ventas.index == dia]
    aux_v1 = aux_v1.values.tolist()
    aux_v2 = [0, 0, 0, 0]

    #Porcentaje venta diario articulos
    aux_a = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    aux_a1 = p_art[p_art.index == dia]
    aux_a1 = aux_a1.values.tolist()
    aux_a2 = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

    #Precios articulos
    aux_p = precios[precios.index == dia]
    aux_p = aux_p.values.tolist()

    respuesta = True
    num_art = 3

    while respuesta == True:

        #Indice del cliente
        cli_ind = np.random.randint(0, len(clientes))
        data['ID_PEDIDO'] = contador
        data['ID_CLIENTE'] = cli_ind + 1
```

```

#Generar fecha del pedido
year = ventas_tiendas['Year'][dia]
month = ventas_tiendas['Month'][dia]
day = ventas_tiendas['Day'][dia]
hora = np.random.randint(0,23)
minuto = np.random.randint(0,59)
segundo = np.random.randint(0,59)
fecha = str(datetime(year, month, day, hora, minuto, segundo))
data['FECHA_PEDIDO'] = fecha

#Indice de la tienda
id_t = np.random.randint(0, len(tiendas))
data['ID_TIENDA'] = id_t + 1

#Indice del articulo
ind_art = 0
precio = True

while precio == True:
    t_art = np.random.randint(0, len(articulos))
    if aux_p[0][t_art] != 0:
        ind_art = t_art
        precio = False

data['ID_ARTICULO'] = ind_art + 1
data['CANTIDAD'] = 1
data['COSTE'] = articulos['COSTE'][ind_art]
data['MARGEN'] = aux_p[0][ind_art] - articulos['COSTE'][ind_art]
data['PRECIO'] = aux_p[0][ind_art]

if aux_v[id_t] < aux_v1[0][id_t]:
    aux_v[id_t] += aux_p[0][ind_art]

if aux_a[ind_art] < aux_a1[0][ind_art]:
    aux_a[ind_art] += aux_p[0][ind_art]
    pedidos.loc[cont, cols] = data
    contador += 1

cont += 1

if aux_a[ind_art] > aux_a1[0][ind_art]:
    aux_a2[ind_art] = 1

if aux_v[id_t] > aux_v1[0][id_t]:
    aux_v2[id_t] = 1

if sum(aux_v2) == len(aux_v2) and sum(aux_a2) == num_art:
    respuesta = False

print("Generando pedidos del dia:" + str(ventas_tiendas['Fecha'][dia]))

pedidos = pedidos.reset_index()
del pedidos['index']

#print(pedidos)
pedidos.to_excel('Pedidos.xlsx', index = False)

```

11.9. Anexo 9: Modelo de predicción ARIMA.


```

#Modelo ARIMA

#Libraries
import pandas as pd
import numpy as np
from matplotlib import pyplot
from statsmodels.tsa.arima_model import ARIMA
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from datetime import datetime, timedelta
from statsmodels.tsa.stattools import adfuller
import math
#from keras import metrics

#split dataset
series = pd.read_excel('Inputs_1.xlsx')
split_point = int(len(series) * 0.8675)
dataset, validation = series[0:split_point], series[split_point:]
print('Train %d, Test %d' % (len(dataset), len(validation)))

# create a differenced series
def difference(dataset, interval=1):
    diff = list()
    for i in range(interval, len(dataset)):
        value = dataset[i] - dataset[i - interval]
        diff.append(value)
    return np.array(diff)

# invert differenced value
def inverse_difference(history, yhat, interval=1):
    return yhat + history[-interval]

def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

# seasonal difference
dataset['UDS_TOTAL'] = dataset['UDS_TOTAL'].astype(float)
X = dataset['UDS_TOTAL'].values
days_in_year = 365
differenced = difference(X, days_in_year)

#Fit the model

p = [3, 5, 7, 10, 12]
d = [0, 1]
q = [1, 2]

data_1 = {'p': [], 'd': [], 'q': [], 'MSE': [], 'MAPE': [], 'MAE': []}
cols_1 = {'p', 'd', 'q', 'MSE', 'MAPE', 'MAE'}

cont = 0

analysis = pd.DataFrame(columns = cols_1)

for i in range(0, len(p)):
    for j in range(0, len(d)):
        for k in range(0, len(q)):

```

```

model = ARIMA(differenced, order=(p[i],d[j],q[k]))
model_fit = model.fit()
# print(model_fit.summary())
# one-step out-of sample forecast
forecast = model_fit.forecast()[0]
# invert the differenced forecast to something usable
forecast = inverse_difference(X, forecast, days_in_year)

# multi-step out-of-sample forecast
num_days_for = 121 #Add number of days and years
forecast_s = model_fit.forecast(steps=num_days_for)[0]
# invert the differenced forecast to something usable
history = [x for x in X]
day = 1

trend = {'FECHA': [], 'UDS_TOTAL': []}
cols = ['FECHA', 'UDS_TOTAL']
data = pd.DataFrame(columns = cols)

ult_date = dataset.loc[len(dataset)-1, 'FECHA']
next_date = ult_date + timedelta(days = 1)

for yhat in forecast_s:
    inverted = inverse_difference(history, yhat, days_in_year)
    trend['UDS_TOTAL'] = int(inverted)
    trend['FECHA'] = next_date
    next_date = next_date + timedelta(days = 1)
    data.loc[day, cols] = trend
    history.append(inverted)
    day += 1

pf = pd.DataFrame(columns = cols)
pf.loc[0, ['FECHA', 'UDS_TOTAL']] = dataset.loc[len(dataset)-1, ...
    ...['FECHA', 'UDS_TOTAL']]
frames = [pf, data]
df = pd.concat(frames)
df = df.reset_index()
del df['index']

MSE = mean_squared_error(validation['UDS_TOTAL'], data['UDS_TOTAL'])
MAE = mean_absolute_error(validation['UDS_TOTAL'], data['UDS_TOTAL'])
MAPE = mean_absolute_percentage_error(validation['UDS_TOTAL'], ...
    ...data['UDS_TOTAL'])

data_1['p'] = p[i]
data_1['d'] = d[j]
data_1['q'] = q[k]
data_1['MSE'] = MSE
data_1['MAPE'] = MAE
data_1['MAE'] = MAE

 analisis.loc[cont, cols_1] = data_1

cont += 1

print('Generando modelo ARIMA: ' + 'p: ' + str(p[i]) + ' d: ' + ...
    ...str(d[j]) + ' q: ' + str(q[k]))

 analisis.to_excel('ARIMA_Analisis.xlsx', index = False)

```

```

 analisis = analisis.sort_values(by = 'MAPE', ascending = True)
 analisis = analisis.reset_index()
 del analisis['index']

 p_opt = int(analisis.loc[0, 'p'])
 d_opt = int(analisis.loc[0, 'd'])
 q_opt = int(analisis.loc[0, 'q'])

 model_opt = ARIMA(differenced, order=(p_opt,d_opt,q_opt))
 model_fit_opt = model_opt.fit()
 print(model_fit_opt.summary())
 # one-step out-of sample forecast
 forecast_opt = model_fit_opt.forecast()[0]
 # invert the differenced forecast to something usable
 forecast_opt = inverse_difference(X, forecast_opt, days_in_year)

 # multi-step out-of-sample forecast
 num_days_for = 121 #Add number of days and years
 forecast_s_opt = model_fit_opt.forecast(steps=num_days_for)[0]
 # invert the differenced forecast to something usable
 history = [x for x in X]
 day = 1

 trend = {'FECHA': [], 'UDS_TOTAL': []}
 cols = ['FECHA', 'UDS_TOTAL']
 data = pd.DataFrame(columns = cols)

 ult_date = dataset.loc[len(dataset)-1, 'FECHA']
 next_date = ult_date + timedelta(days = 1)

 for yhat in forecast_s:
     inverted = inverse_difference(history, yhat, days_in_year)
     trend['UDS_TOTAL'] = int(inverted)
     trend['FECHA'] = next_date
     next_date = next_date + timedelta(days = 1)
     data.loc[day,cols] = trend
     history.append(inverted)
     day += 1

 pf = pd.DataFrame(columns = cols)
 pf.loc[0,['FECHA', 'UDS_TOTAL']] = dataset.loc[len(dataset)-1,['FECHA', 'UDS_TOTAL']]
 frames = [pf, data]
 df = pd.concat(frames)
 df = df.reset_index()
 del df['index']

 fig = pyplot.figure()
 pyplot.plot(dataset['FECHA'],dataset['UDS_TOTAL'], color = 'blue',...
             ...label = 'Time Serie')
 pyplot.plot(df['FECHA'], df['UDS_TOTAL'], color = 'red', label = 'Forecast')
 pyplot.xticks(rotation = 25)
 pyplot.legend()
 pyplot.xlabel('Fecha', fontsize = 10)
 pyplot.ylabel('Uds Vendidas', fontsize = 10)
 pyplot.show()

 fig.savefig('Forecast_v2.jpg', dpi=100)
 df.to_excel('Result_forecast_v2.xlsx', index = False)

```


11.10. Anexo 10: Modelo de predicción FbProphet.


```

#Modelo FBProphet

#Libraries
import logging
logging.getLogger('fbprophet').setLevel(logging.ERROR)
import warnings
warnings.filterwarnings("ignore")

import pandas as pd
import numpy as np
from fbprophet import Prophet
from sklearn.metrics import mean_squared_error

#Quick Start
df = pd.read_csv('Inputs_2.csv')
split_point = int(len(df)*0.8675)
df_1, df_2 = df[0:split_point], df[split_point:]

#Holidays definition

vacations = pd.DataFrame({'holiday': 'Holiday', 'ds': pd.to_datetime(...
...list(pd.read_csv('Holidays.csv')['ds']))})
black_friday = pd.DataFrame({'holiday': 'BlackFriday', 'ds': pd.to_datetime(...
...list(pd.read_csv('BlackFriday.csv')['ds']))})
promotions = pd.DataFrame({'holiday': 'Promotion', 'ds': pd.to_datetime(...
...list(pd.read_csv('Promotions.csv')['ds']))})
raining = pd.DataFrame({'holiday': 'Raining', 'ds': pd.to_datetime(...
...list(pd.read_csv('Raining.csv')['ds']))})
lowtemperature = pd.DataFrame({'holiday': 'LowTemperature', 'ds': pd.to_datetime(...
...list(pd.read_csv('Temperature.csv')['ds']))})

all_events = pd.concat((vacations, black_friday, promotions, raining,...
...lowtemperature))
all_events = all_events.sort_values(['ds'])

#Parameters improvement process

s = [2, 5, 10, 12, 15]
c = [0.05, 0.075, 0.1, 0.25, 0.5]
h = [0.5, 0.75, 1, 2, 5]

cont = 0
dt = {'Seasonality': [], 'Changepoint': [], 'Holidays': [], 'MSE_train': [],...
... 'MAE_train': [], 'MAPE_train': [], 'MSE_test': [], 'MAE_test': [],...
... 'MAPE_test': [], 'MASE_train': [], 'MASE_test': []}
cols = {'Seasonality', 'Changepoint', 'Holidays', 'MSE_train', 'MAE_train',...
... 'MAPE_train', 'MSE_test', 'MAE_test', 'MAPE_test', 'MASE_train', 'MASE_test'}
analysis = pd.DataFrame(columns = cols)

def make_comparison_dataframe(historical, forecast):

    return forecast.set_index('ds')[['yhat', 'yhat_lower', 'yhat_upper']].join(...
...historical.set_index('ds'))

def calculate_forecast_errors(df, prediction_size):
    df = df.copy()

    df['e'] = df['y'] - df['yhat']
    df['p'] = 100 * df['e'] / df['y']

```

```

predicted_part = df[-prediction_size:]

error_mean = lambda error_name: np.mean(np.abs(predicted_part[error_name]))

return error_mean('p'), error_mean('e')

def calculate_MASE(value_set, prediction_set):

    e = np.abs(value_set - prediction_set)
    n = value_set.shape[0]
    d = np.diff(value_set).sum() / (n-1)

    error = np.abs(e / d).mean()

    return error

for i in range(0, len(s)):
    for j in range(0, len(c)):
        for k in range(0, len(h)):

            m = Prophet(holidays = all_events, seasonality_prior_scale = s[i],...
                        ...changeoint_prior_scale = c[j], holidays_prior_scale = h[k])
            m.fit(df_1)

            future = m.make_future_dataframe(periods = 121, freq = 'D',...
                        ...include_history = True)

            forecast = m.predict(future)
            forecast_1 = forecast[0:split_point]
            forecast_2 = forecast[split_point:]

            MSE_train = mean_squared_error(df_1['y'], forecast_1['yhat'])
            MSE_test = mean_squared_error(df_2['y'], forecast_2['yhat'])

            cmp_df1 = make_comparison_dataframe(df_1, forecast_1)
            cmp_df2 = make_comparison_dataframe(df_2, forecast_2)

            MAE_train, MAPE_train = calculate_forecast_errors(cmp_df1, 792)
            MAE_test, MAPE_test = calculate_forecast_errors(cmp_df2, 121)

            MASE_train = calculate_MASE(df_1['y'], forecast_1['yhat'])
            MASE_test = calculate_MASE(df_2['y'], forecast_2['yhat'])

            dt['Seasonality'] = s[i]
            dt['Changeoint'] = c[j]
            dt['Holidays'] = h[k]
            dt['MSE_train'] = MSE_train
            dt['MAE_train'] = MAE_train
            dt['MAPE_train'] = MAPE_train
            dt['MSE_test'] = MSE_test
            dt['MAE_test'] = MAE_test
            dt['MAPE_test'] = MAPE_test
            dt['MASE_train'] = MASE_train
            dt['MASE_test'] = MASE_test

            analysis.loc[cont, cols] = dt

            cont += 1

```

```

        print('Generando análisis para la combinación: ' + 'Seasonality: ' + ...
        ...str(s[i]) + ', Changeoint: ' + str(c[j]) + ', Holidays: ' + str(h[k]))

#print(analysis)

analysis = analysis.sort_values(by = 'MASE_test', ascending = True)
analysis = analysis.reset_index()
del analysis['index']

analysis.to_csv('FB_Analysis.csv', index = False)

seasonality_prior_scale_opt = analysis.loc[0, 'Seasonality']
changepoint_prior_scale_opt = analysis.loc[0, 'Changeoint']
holidays_prior_scale_opt = analysis.loc[0, 'Holidays']

#print(seasonality_prior_scale_opt)
#print(changepoint_prior_scale_opt)
#print(holidays_prior_scale_opt)

m_opt = Prophet(holidays = all_events, seasonality_prior_scale = ...
...seasonality_prior_scale_opt, changepoint_prior_scale = ...
...changepoint_prior_scale_opt, holidays_prior_scale = holidays_prior_scale_opt)
m_opt.fit(df_1)

future_opt = m_opt.make_future_dataframe(periods = 121, freq = 'D', ...
...include_history = True)

forecast_opt = m_opt.predict(future_opt)

forecast_opt.to_csv('Forecast.csv', index = False)
fig_1 = m_opt.plot(forecast_opt)
fig_1.savefig('Forecast_fb.png', dpi=100)
fig_2 = m.plot_components(forecast_opt, uncertainty = False)
fig_2.savefig('Forecast_fb_comp.png', dpi=100)

```

