



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA DE ORGANIZACIÓN INDUSTRIAL

DESARROLLO Y ANÁLISIS DE UNA HERRAMIENTA DE CÁLCULO DE RUTAS DE TRANSPORTE A CAPACIDAD FINITA PARA ABASTECER A UNA LÍNEA DE MONTAJE DE AUTOMÓVILES

AUTORA: LORENA BAVIERA FORNER

TUTOR: CARLOS ANDRÉS ROMANO

Curso académico: 2015-16

RESUMEN

El objetivo del presente Trabajo Final de Grado (TFG) es configurar un programa informático que permita minimizar los costes de transporte a las líneas en la planta de montaje de Ford Almussafes (Valencia). El programa informático tratará de encontrar las mejores rutas posibles para suministrar a los distintos puestos de trabajo en la línea de montaje.

Ford es una empresa con una estrategia de bajo costo, es decir, ofrece precios bajos al consumidor para estimular la demanda y crecer en cuota de mercado. Para conseguirlo, el primer sitio donde debe haber bajos precios es dentro de la empresa, con lo que se tendrán que minimizar los costes durante todo el proceso productivo y este TFG presenta esa oportunidad de lograrlo.

Para alcanzar este objetivo, se ha hecho uso de un modelo implementado en el entorno MPL (Mathematical Programming Language) y resuelto mediante el optimizador COINMP y de una herramienta informática utilizando *Visual Basic for Applications*. Mediante éstas, se calculan las mejores rutas posibles que deberán seguir las distintas carretillas suministradoras de la planta, así como el número de éstas que son necesarias para abastecer las necesidades de cada puesto de la línea de montaje.

SUMMARY

The main objective of the present “TFG” (Trabajo Final de Grado) is to set up a software tool that will enable to minimize transport costs of the lines in the assembly plant of Ford Almussafes (Valencia). This software tool tries to find the best possible routes to supply the different workstations in the assembly line.

Ford is a company that follows a low cost strategy, in other words, it offers low prices on its products for the consumer to stimulate the demand and increase its market share. In order to succeed, the first place where there should be low prices is inside the company, therefore, all costs during the productive process should be minimized and this TFG offers the opportunity to achieve it.

To reach this goal, a model has been implemented in the MPL (*Mathematical Programming Language*) context and solved by means of the optimizer *COINMP* and a software tool has been developed making use of *Visual Basic for Applications* (VBA). By means of both, the best possible routes that the delivery forklift trucks must follow in the plant will be calculated, as well as the number of these that are required to fulfil the necessities of each workstation in the assembly line.

RESUM

L'objectiu del present Treball Final de Grau (TFG) es configurar un programa informàtic que permeti minimitzar els costos de transport a les línies de la planta de muntatge de Ford Almussafes (València). El programa informàtic tractarà de trobar les millors rutes possibles per a subministrar els diferents llocs de treball en la línia de muntatge.

Ford és una empresa amb una estratègia de baixos costos, és a dir, ofereix baixos preus al consumidor per a estimular la demanda i créixer en quota de mercat. Per a aconseguir-ho, el primer lloc on ha d'haver-hi baixos preus es a l'empresa, per la qual cosa s'hauran de minimitzar els costos durant tot el procés productiu i aquest TFG presenta aqueixa oportunitat d'aconseguir-ho.

Per a aconseguir aquest objectiu, s'ha fet ús d'un model implementat en l'entorn MPL (*Mathematical Programming Language*) i resolt mitjançant l'optimitzador *CoinMP* i d'una eina informàtica utilitzant *Visual Basic for Applications*. Mitjançant aquestes, es calculen les millors rutes possibles que hauran de seguir els diferents carretons subministradors de la planta, així com el nombre d'aquests que són necessaris per a proveir les necessitats de cada lloc de la línia de muntatge.

ÍNDICE GENERAL

MEMORIA.....	8
1. OBJETO, MOTIVACIÓN Y JUSTIFICACIÓN DEL PROYECTO	
1.1. Objeto.....	10
1.2. Motivación.....	10
1.3. Justificación.....	10
2. ANTECEDENTES	
2.1. Descripción de la Empresa.....	12
2.2. Logística y Logística Interna	
2.2.1. <i>Cadena de Suministro</i>	18
2.2.2. <i>Estrategia de Planificación Agregada</i>	20
2.2.3. <i>Lean Manufacturing</i>	21
2.2.4. <i>Milk-run</i>	22
3. ANÁLISIS DE LA SITUACIÓN INICIAL	
3.1. Análisis general del Proceso de Abastecimiento.....	26
3.2. Análisis de los dispositivos de Manutención.....	26
3.3. Descripción detallada del Proceso de Abastecimiento.....	27
4. PROPUESTA DE UN MODELO MATEMÁTICO PARA SU RESOLUCIÓN	
4.1. Conceptos de la Modelización Matemática	
4.1.1 <i>Modelización Matemática</i>	30
4.1.2 <i>Resolución Informática (MPL)</i>	34
4.2. Planteamiento de la Situación Inicial.....	35
4.3. Formulación del Modelo.....	37
4.4. Resolución del Modelo.....	43
4.5. Interpretación de los Resultados.....	44
4.6. Evaluación de los Resultados.....	46
5. PROPUESTA DE UNA HERRAMIENTA DE CÁLCULO DE RUTAS DE TRANSPORTE	
5.1. Visual Basic for Applications (VBA)	
5.1.1 <i>Definición VBA</i>	48
5.1.2 <i>Procedimiento Inicial</i>	48
5.1.3 <i>Características de la herramienta</i>	50
5.2. Algoritmos utilizados.....	53
5.3. Análisis de requisitos.....	54
5.4. Diseño del Programa y Codificación.....	54
5.5. Pruebas y Verificación.....	62
6. CONCLUSIONES FINALES Y RECOMENDACIONES.....	77
7. BIBLIOGRAFÍA.....	79
8. ANEXOS DE LA MEMORIA	
8.1 Modelo Matemático.....	80
8.2 Herramienta informática.....	82
8.3 Diagrama de Flujo del Programa.....	93
8.4 Índices de Imágenes, Diagramas, Tablas y Gráficas.....	97

ÍNDICE GENERAL

PRESUPUESTO.....	102
9. PRESUPUESTO	
9.1. Presupuesto de Inversión.....	104
9.2. Presupuesto de Explotación.....	105
9.3. Rentabilidad Económica.....	107
9.4. Conclusiones.....	110
10. ANEXOS DEL PRESUPUESTO.....	111



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

MEMORIA

DESARROLLO Y ANÁLISIS DE UNA HERRAMIENTA DE CÁLCULO DE RUTAS DE TRANSPORTE A CAPACIDAD FINITA PARA ABASTECER A UNA LÍNEA DE MONTAJE DE AUTOMÓVILES

AUTORA: LORENA BAVIERA FORNER

TUTOR: CARLOS ANDRÉS ROMANO

Curso académico: 2015-16

ÍNDICE MEMORIA

1. OBJETO, MOTIVACIÓN Y JUSTIFICACIÓN DEL PROYECTO	
1.1. Objeto.....	10
1.2. Motivación.....	10
1.3. Justificación.....	10
2. ANTECEDENTES	
2.1. Descripción de la Empresa.....	12
2.2. Logística y Logística Interna	
2.2.1. <i>Cadena de Suministro</i>	18
2.2.2. <i>Estrategia de Planificación Agregada</i>	20
2.2.3. <i>Lean Manufacturing</i>	21
2.2.4. <i>Milk-run</i>	22
3. ANÁLISIS DE LA SITUACIÓN INICIAL	
3.1. Análisis general del Proceso de Abastecimiento.....	26
3.2. Análisis de los dispositivos de Manutención.....	26
3.3. Descripción detallada del Proceso de Abastecimiento.....	27
4. PROPUESTA DE UN MODELO MATEMÁTICO PARA SU RESOLUCIÓN	
4.1. Conceptos de la Modelización Matemática	
4.1.1. <i>Modelización Matemática</i>	30
4.1.2. <i>Resolución Informática (MPL)</i>	34
4.2. Planteamiento de la Situación Inicial.....	35
4.3. Formulación del Modelo.....	37
4.4. Resolución del Modelo.....	43
4.5. Interpretación de los Resultados.....	44
4.6. Evaluación de los Resultados.....	46
5. PROPUESTA DE UNA HERRAMIENTA DE CÁLCULO DE RUTAS DE TRANSPORTE	
5.1. Visual Basic for Applications (VBA)	
5.1.1. <i>Definición VBA</i>	48
5.1.2. <i>Procedimiento Inicial</i>	48
5.1.3. <i>Características de la herramienta</i>	50
5.2. Algoritmos utilizados.....	53
5.3. Análisis de requisitos.....	54
5.4. Diseño del Programa y Codificación.....	54
5.5. Pruebas y Verificación.....	62
6. CONCLUSIONES FINALES Y RECOMENDACIONES.....	77
7. BIBLIOGRAFÍA.....	79
8. ANEXOS DE LA MEMORIA	
8.1 Modelo Matemático.....	80
8.2 Herramienta informática.....	82
8.3 Diagrama de Flujo del Programa.....	93
8.4 Índices de Imágenes, Diagramas, Tablas y Gráficas.....	97

1. OBJETO, MOTIVACIÓN Y JUSTIFICACIÓN DEL PROYECTO

1.1 Objeto

El objeto de este Trabajo Final de Grado, conocido por las siglas “TFG”, es desarrollar un programa informático que permita minimizar los costes en cuanto al suministro de las distintas piezas a cada puesto de la línea en la planta de montaje de Ford Almussafes. Esta planta consta de muchas líneas de montaje y en cada una de ellas se realizan distintas tareas con diversas piezas. Para poder saber cómo y cuándo hay que servir a cada línea, mediante MPL se creará un modelo matemático y con VBA se desarrollará un algoritmo que resolverá la situación planteada.

1.2 Motivación

En cuanto a la motivación académico personal, este proyecto permite al alumno profundizar en el uso de dos herramientas informáticas que se han visto durante la carrera: *Mathematical Programming Language* y *Visual Basic for Applications*. Con su utilización se consigue una mayor soltura, rapidez y entendimiento a la hora de modelar los problemas que aparecen en Ingeniería de Organización Industrial y programarlos para poder resolverlos de forma rápida. Además, este trabajo da la oportunidad de conocer mejor cómo funciona realmente la logística interna de la empresa Ford.

1.3 Justificación

Este TFG permite al alumno poner en práctica conocimientos que se han ido adquiriendo durante el Grado en Ingeniería de Organización Industrial. Estas competencias se han obtenido de asignaturas como “Planificación de la Producción e Inventarios”, “Métodos Cuantitativos” y “Programación y Control de Producción y Operaciones”. Estas tres asignaturas dan la oportunidad al alumno de poder optimizar los recursos y minimizar los costes.

Actualmente, para ser una fuerza competitiva en el sector del automóvil, las empresas pueden seguir fundamentalmente dos estrategias. Una de ellas es la estrategia de diferenciación, ofreciendo automóviles con elementos singulares y diferenciadores, bien en el diseño, la equipación, los elementos de seguridad, el motor, la propia conducción, etc. Por ejemplo, Audi sigue una estrategia de diferenciación, mientras que en el caso de Ford, éste sigue la otra estrategia, la estrategia de bajos precios. En Ford se ha optado por una estrategia de ofertar automóviles a bajos precios de venta y por tanto, cuando se quiera maximizar el beneficio empresarial habrá que centrar los esfuerzos en los métodos que permitan reducir los costes de producción.

Una empresa enfocada en minimizar costes, pero sin disminuir la calidad de los materiales utilizados, básicamente puede actuar en dos frentes: por un lado se precisa eliminar del proceso productivo aquellos elementos prescindibles que realmente no producen valor añadido para el cliente y por otro lado, intentar ahorrar tiempo en cada una de las fases del proceso productivo, ya que el ahorrar tiempo se traducirá en reducir costes. Cuando una compañía lleva tantos años siguiendo esta misma estrategia, cabría pensar que se llegará a un punto donde será imposible seguir ahorrando en tiempos de producción.

La motivación para la realización de este trabajo ha estado precisamente en la dificultad mencionada, escogiendo una sección del proceso productivo sobre el que todavía no se había incidido, concretamente en el modelo de suministro de piezas a cada puesto de trabajo de cada línea en la planta de montaje.

Cada línea de montaje requiere de diversas piezas para hacer sus tareas, y necesita que éstas estén en cada puesto de la línea justo en el momento en que se demanden puesto que si las piezas llegaran antes de tiempo, podría suponer dos problemas: el primero, un aumento en los costes de inventariar puesto que se está almacenando un material que no es necesario y el segundo, un colapso debido al limitado espacio con que cuenta la línea para recepción de materiales. Si por el contrario se suministrase tarde, esto podría significar una parada de la línea por insuficiencia de material y la insatisfacción del cliente por no servirle a tiempo. Por todo esto, es muy importante gestionar adecuadamente la llegada de las piezas a cada línea. Además de saber cuándo se suministrarán las piezas a las distintas líneas, se intentará encontrar el menor número de carretillas necesarias para satisfacer la demanda de todas las líneas.

Conviene aclarar que este programa informático que se ha desarrollado tiene aplicación no sólo para la compañía modelo escogida, Ford Almussafes, sino que es aplicable para cualquier empresa que cuente con un sistema productivo donde esté identificado un punto de salida y varios puntos a los que se tiene que suministrar.

Finalmente, indicar que modificando ligeramente el programa, también podría utilizarse para estudiar la situación contraria, es decir, para casos en que se parte de varios puntos de salida y se tiene un punto de entrada. Este sería el caso de prácticamente cualquier empresa del sector de la automoción, puesto que generalmente cuentan con varios proveedores (varios puntos de salida) y que les proveen a ellos (un punto de entrada). Aquí se trataría también de minimizar los costes, pero en este caso serían los costes de suministrar a la empresa.

2. ANTECEDENTES

2.1 Descripción de la Empresa

Ford tuvo sus inicios en el sector automovilístico alrededor de 1903 en Detroit, Estados Unidos. La empresa fue creada por Henry Ford. Al comienzo se producían pocos coches por día debido a la baja demanda y al pequeño tamaño de la empresa. En ella, grupos de dos o tres hombres trabajaban en cada automóvil usando piezas fabricadas por diferentes proveedores y ensamblándolas todas. La idea que se utilizaba al principio era la de una unidad estática de montaje, donde eran los operarios los que se acercaban al vehículo para poder montarlo.



Imagen 1. Imagen de 1910 donde se observa que el vehículo está estático y son los operarios los que se mueven para realizarle las operaciones necesarias. (Imagen extraída de *Henry Ford, Mi vida y obra, 1925.*)

En muy poco tiempo, la demanda de los clientes aumentó radicalmente, y la empresa tuvo que trasladarse a unas instalaciones más grandes para poder satisfacerla. En esos momentos la empresa había desarrollado diversas técnicas de producción, las cuales permitían reducir los tiempos de ensamblaje de los automóviles. Se pasó de tardar 12h en fabricar una unidad, a producir 1 unidad por hora. Estas nuevas técnicas de producción consistían en cintas transportadoras donde se colocaba el coche y los operarios actuaban sobre él y no al contrario, como se había hecho hasta entonces. Desde aquel momento y hasta la actualidad, Ford ha ido creciendo y cuenta con plantas de ensamblaje en muchos países del mundo, creando automóviles ajustados al gusto del consumidor y produciendo de la misma manera.



Imagen 2. Línea de ensamblaje en Ford. Los operarios siempre están en la misma posición y son los vehículos los que están en movimiento mediante una cinta transportadora. (Imagen extraída de Alvolante info, 2016.)

Una de las plantas de Ford se encuentra en Valencia, de donde se han extraído los datos para la realización del presente trabajo.

Desde 1974 se encuentra en Almussafes (Valencia) la planta que tiene Ford en España. Aunque la fábrica se construyera en 1974, no fue hasta 1975 cuando se fabricó el primer motor y un año después se construyese el primer modelo. Este modelo fue el Ford Fiesta, que a día de hoy sigue en el mercado y teniendo mucho éxito, aunque ha sufrido variaciones desde su comienzo. La planta cuenta actualmente con 8000 empleados y con una superficie de 270 hectáreas.



Imagen 3. Vista de toda la planta de Ford Almussafes. (Imagen extraída de Autobild.es, 2016.)

Con el paso de los años, la empresa empezó a producir más modelos de coches y actualmente cuenta con los siguientes: Mondeo, New Kuga, Transist Connect, además de la incorporación de dos nuevos modelos este año (2015): S-Max y Galaxy.

La construcción de los nuevos modelos ha conllevado la instalación de 900 nuevos robots desde hace cuatro años y que aumenten las necesidades de suministro a 18.000 componentes, frente a las 7.000 piezas que utilizaba Ford Almussafes antes de ampliar su carga de trabajo.

A pesar de la incorporación de dos nuevos modelos, que son más complejos que lo que se había estado produciendo anteriormente (como es el Ford Focus), Ford Almussafes conseguir producir un coche cada 80 segundos. Actualmente la capacidad máxima de Ford es de 450.000 unidades/año, no obstante, en 2015 se preveía acabar produciendo 400.000 unidades. La fabricación de estas 400.000 unidades implica un aumento de producción del 40% respecto al año anterior. Además, el hecho de combinar cinco modelos, demuestra que actualmente la planta de Almussafes sea la más flexible de toda Europa.

El complejo productivo de Ford en Almussafes se puede dividir en cuatro grandes áreas:

1. Los departamentos de Servicios. Esta área está formada por el centro de formación, las oficinas centrales, instalaciones deportivas para todos los miembros de Ford y otros servicios auxiliares.

2. La planta de Recambios. Es un almacén situado en la zona norte de la fábrica Ford, en la que como su nombre indica, almacena piezas de recambios. Estas piezas de recambios servirán para abastecer el mercado nacional.
3. La planta de Motores. En esta planta se realizan el mecanizado de piezas y el montaje de motores. En esta planta se producen motores para otros fabricantes de automóviles como por ejemplo Jaguar, Land Rover y Volvo.
4. Área de Fabricación. En esta última área, se realiza la fabricación final de los automóviles. En el proceso de fabricación del automóvil, éste tiene que pasar por varias secciones en el orden siguiente:

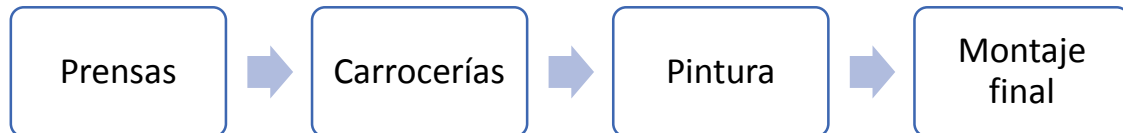


Diagrama 1. Representación de las diferentes secciones que se encuentran en el área de Fabricación.

Una vez conocidas las diferentes áreas en las que se subdivide el área de fabricación, se procederá a conocer con más detalle cada una de éstas.

Área de Prensas

La factoría de Almussafes cuenta con aproximadamente 50 prensas que se utilizan para fabricar algunas piezas para los automóviles (no todas las piezas se fabrican en la planta ya que hay otras piezas que se subcontratan). Estas prensas están distribuidas a lo largo de las 9 líneas. Además de estas prensas tradicionales, la planta de Almussafes también cuenta con tres prensas triaxiales.

El primer proceso en la planta de Prensas es el de cortar las bobinas de acero que llegan a la planta en piezas más pequeñas. Después, una máquina hidráulica corta las piezas en el tamaño requerido. Estas piezas se utilizarán más tarde en la carrocería. Un ejemplo de estas piezas son los refuerzos para los techos. Éstos, se producen a una velocidad de 20 piezas/min, en cambio, otras piezas más pequeñas como podrían ser las puertas, se cortan 40 piezas/min.

Una vez las piezas están cortadas en sus determinados tamaños, éstas son llevadas en “palets” a la zona de estampación para darles la forma correspondiente. Para llevarlo a cabo se utilizan prensas con potencia de 11.000 toneladas. En contra de lo que podría pensarse, estas prensas no moldean por tener una alta temperatura (están a temperatura ambiente), sino que se adquiere la forma de la pieza mediante el peso de la prensa.

Las prensas están dotadas de un sistema de célula fotoeléctrica que sirve para detectar cualquier presencia humana u objeto. Al detectarla, la prensa se bloquea, evitando así cualquier daño accidental. Estas prensas reciben un mantenimiento exhaustivo de carácter preventivo: se realiza un mantenimiento periódico para que las máquinas no lleguen a fallar, pues el fallo de una de estas prensas significa en muchas ocasiones una parada de la máquina y esto supone un coste importante. Aun así, en ocasiones hay que aplicar mantenimiento correctivo para reparar su funcionamiento defectuoso, actuando de la forma más rápida posible para que el tiempo de parada de la línea sea el menor posible.

En el área de Prensas se está volviendo a trabajar como se empezó a trabajar al inicio de la creación de la fábrica, que es realizando más mantenimiento preventivo que correctivo. Esto es lo correcto pues como se ha podido ver, es muy perjudicial para la empresa el tener máquinas paradas. Se debería ir aplicando continuamente el mantenimiento preventivo pues esto evita o hace que disminuyan las posibilidades de que una prensa deje de funcionar y haya que aplicarle mantenimiento correctivo. Ford había dejado de aplicar mantenimiento preventivo ya que supone un mayor esfuerzo y coste a corto plazo el aplicar más mantenimiento preventivo que correctivo, aunque a largo plazo, siempre es mucho más beneficioso aplicar el menor mantenimiento correctivo posible.

Todos los procesos existentes en la planta de Prensas están muy automatizados, a excepción de la verificación de la calidad de las piezas. Este proceso debe realizarlo un operario ya que hasta el momento no se cuenta con un robot que sea capaz de distinguir defectos.

En cuanto a la seguridad, en esta área se deben utilizar tapones para los oídos ya que el ruido existente en la planta es mayor de 80 dB. Además, se debe contar siempre con guantes dado que el material con el que se trata puede producir cortes en el personal.

Área de Carrocerías

Esta sección es la encargada de fabricar y ensamblar varias piezas de la carrocería. Ambos procesos se consiguen con un tiempo de ciclo de 58 segundos. Para fabricar estas piezas existen dos líneas diferentes, donde cada una de ellas está destinada a unos modelos concretos de automóviles.

En el siguiente diagrama pueden verse los modelos que son fabricados en cada línea:

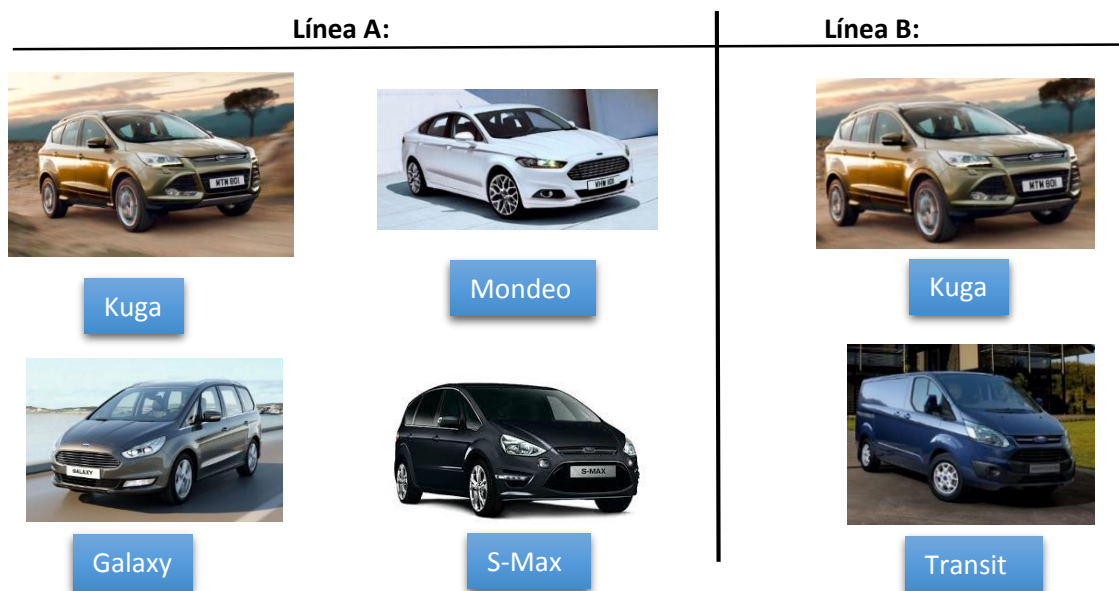


Diagrama 2. Representación de los diferentes modelos que se producen en cada línea.

La sección de carrocerías es la primera área desde que se ha iniciado el proceso de producción donde el futuro vehículo va a empezar a tomar rutas distintas de producción según el modelo que finalmente vaya a producirse. Para saber por qué se produce cada modelo en cada línea, hay que mirar las cantidades que se producen de cada modelo.

Modelo	Producciones aproximadas / día	% Producción total
New Kuga	2000	23,53
Transit	3000	35,29
Mondeo	2000	23,53
S-max	1000	11,76
Galaxy	500	5,89
Total	8500	100,00

Tabla 1. Producciones aproximadas/modelo y día. (Datos proporcionados por Ford.)

Mediante esta tabla y el diagrama superior se puede ver cómo actualmente hay un aceptable equilibrio entre ambas líneas ya que si sólo se utilizasen las producciones de Mondeo, S-max y Galaxy en una línea, se llegaría a cubrir solamente un 41,18% de la producción diaria y mediante la línea 2, el restante, 58,82%. Por lo tanto, y como puede comprobarse en el diagrama anterior, el modelo New Kuga es producido en ambas líneas para ir compensando la línea que tenga un menor porcentaje de producción.

En esta sección se trabaja tanto con robots terrestres como aéreos. Esto es de gran ayuda ya que se equilibra el tiempo de ciclo y además se evitan congestiones por parte de los robots en los pasillos. Estos robots se dedican a la unión entre suelo y compartimento del motor. Todo esto también implica un menor número de accidentes y que se pueda transportar más material dado que no sólo se transporta en el espacio disponible, sino también por el aire.

Cuando los vehículos llegan al final de las líneas de carrocería, éstos se almacenan hasta que se lleve a cabo su secuenciación para entrar posteriormente en el área de pintura, a la cual se accede a través de un ascensor y un túnel aéreo.

Pintura

Esta zona es la que está más controlada en cuanto al acceso de personal externo puesto que es muy sensible y el producto se podría contaminar muy fácilmente. Todo personal autorizado debe llevar una bata y guantes, y antes de entrar en la planta existe un soplador de aire que elimina el polvo adherido a la ropa para que éste no contamine más tarde la pintura.

La planta está dividida en dos líneas de pintura. Esto se debe a que cada una se dedica a la pintura de diferentes modelos. Cada línea contiene los mismos modelos que en las dos líneas de la sección de Carrocería.

- Pintura A: Kuga, Mondeo, Galaxy y S-Max
- Pintura B: Kuga y Transit

En primer lugar, se procede a un lavado y desengrasado, a lo que sigue la fase de fosfatación. Este tratamiento consiste en aplicar una protección anticorrosiva a toda la carrocería. Su objetivo es que el color de la carrocería permanezca intacto durante muchos años.

Una vez pasado el proceso de fosfatación, la carrocería se sumerge en un baño de cataforesis, en el que por medio de un campo eléctrico, la pintura queda adherida. Esta primera capa de pintura que se aplica es de color gris, sea cual sea el modelo y el color final de éste. Una vez pintada con esta primera capa, se procede a pintar el vehículo con el color exigido por el cliente.

Más tarde, una laca transparente es aplicada a la carrocería para darle brillo. Además, cuenta con una protección ultravioleta que evita que el sol ataque el color del automóvil. Después de esto, la carrocería está lista para secarse. Para ello se utiliza un horno de secada a 150 grados.

Una vez realizado el proceso de pintura, las puertas son separadas del resto del vehículo justo antes de entrar en el proceso de Montaje.

Montaje

La planta de Montaje es la última sección y se encarga del interior del coche. En ella deben añadirse y ensamblarse todas las piezas que componen el automóvil hasta que el coche esté completo. Entre otras, estas piezas pueden ser ruedas, motores, puertas y demás accesorios puesto que después de aquí ya debe salir un vehículo disponible para los usuarios finales.

Esta sección se puede dividir en cuatro procesos:

- 1) Línea de Trim
- 2) Línea de Pre-Chasis
- 3) Línea de Chasis
- 4) Zona de pruebas

1) Línea de Trim:

Esta línea a su vez se subdivide en dos líneas diferentes dependiendo de cada modelo:

- Trim A: Kuga, Mondeo, Galaxy y S-Max
- Trim B: Kuga y Transit

En la línea de Trim se realizan todos los pre-montajes iniciales, como son: conjunto del salpicadero, circuito eléctrico, circuito de calefacción, asientos, guarnecido interior, airbag, volante, cristales y faros. Cada operación es realizada por un operario distinto y es la carrocería la que se mueve, no el operario ya que así los tiempos se ven reducidos. Como se ha mencionado anteriormente, esta metodología lleva introducida desde los inicios de Ford.

2) Línea de Pre-Chasis

La función de esta línea es añadir las puertas. Como se ha mencionado anteriormente, durante todo el proceso hasta llegar a la planta de Montaje, el vehículo contaba con las puertas, pero, al entrar en esta planta éstas son separadas del resto del coche para que los operarios tengan una mayor facilidad a la hora de realizar su trabajo. Si se mantuvieran las puertas sería muy difícil para el operario introducir el conjunto de piezas en el interior del coche. Además, conllevaría el riesgo de que éstas pudieran dañarse mientras se realizan dichas operaciones.

3) Línea de Chasis

Las operaciones en esta línea se realizan con la carrocería suspendida en el aire, siendo colgadas por un transportador aéreo de tres vías. Aquí es dónde se añaden inicialmente suspensiones, ejes, transmisión y el conjunto del motor. El ensamblaje del motor con el eje trasero y el tubo de escape es una de las pocas operaciones que deben realizarse manualmente, pues se requiere de una gran precisión.

Además de esto, durante el resto de la línea se van añadiendo al vehículo otros elementos como son el radiador, los depósitos de líquidos y por último, las pastillas de freno y el conjunto del tubo de escape.

4) Zona de Pruebas

Esta zona se utiliza para practicar diversas pruebas al que será el vehículo final. Estas pruebas incluyen: pruebas de aceleración, de alineación, de estanqueidad. Una vez hechas y superadas, el vehículo pasará a la línea de aceptación final. Pero, antes de salir, se le aplica una capa de cera para protegerlo de camino a su destino.

Una vez terminado el proceso de montaje, el vehículo ya está listo para abandonar la fábrica y ser puesto a la venta.

Es evidente que Ford Almussafes es una planta de grandes dimensiones y con una amplia gama de modelos en producción. Prueba de ello es que desde hace unos meses se ha convertido en la principal planta de Ford en Europa, exportando a 75 países. Una parte importante del éxito de la compañía depende de una buena gestión de la planta en cuanto a recursos y logística.

Como este TFG quería centrarse en un área de la que Ford pudiera beneficiarse, se ha decidido estudiar la planta de Montaje final, en la que la logística está muy presente. Esta área conlleva muchas operaciones de ensamblaje, y por tanto, el correcto abastecimiento de todas las piezas que deben añadirse en cada puesto de trabajo es un factor primordial y cualquier ahorro encontrado en el suministro supondrá una gran disminución en los costes de la empresa.

2.2 Logística y Logística Interna

2.2.1 Cadena de Suministro

Según *“The Council of Logistics Management”*, se puede definir la logística como: “el proceso de planificar, implementar y controlar eficientemente el flujo de materias primas, productos en curso, productos terminados y la información relacionada con ellos, desde el punto de origen hasta el punto de consumo, con el propósito de satisfacer los requerimientos del cliente.”

De la definición anterior se puede extraer que una buena implementación de la logística beneficiará y ayudará a que la empresa tenga éxito, pues se encarga de todo el movimiento de productos e información, desde el inicio del proceso hasta el cliente. Una buena gestión de éstos implicaría también que se minimizaran costes. Esto se puede ver con el siguiente ejemplo. Si existe un buen flujo de información entre cliente y fabricante, este último podrá saber cuánto debe producir. Éste fabricará justo lo que el cliente le pide y por lo tanto, no tendrá muchas unidades para inventariar, lo que minimizará los costes.

Añadiendo a lo anterior, si se desea tener una buena gestión de la logística, hay que involucrar a todos los participantes de la cadena de suministro. En la cadena de suministro se encuentra el conjunto de empresas que contribuyen a la creación, distribución, manipulación, almacenamiento y comercialización de un producto.

A continuación se puede observar una típica cadena de suministro que suele representar a muchas empresas:



Diagrama 3. Representación de una cadena de suministro.

Además, hay que recalcar que durante toda la cadena de suministro debe haber comunicación y un flujo de información entre todos los participantes de la misma.

Anteriormente, cada empresa trabajaba de manera independiente, con lo que no había un flujo de comunicación entre los distintos miembros de la cadena de suministro. Esto suponía un grave inconveniente puesto que al carecer de información podría ocurrir tanto que la empresa produjera menos unidades de las que demandaba el cliente, como al contrario, que produjera en exceso y se tuvieran unidades inventariadas. En ambos casos, esto incrementaba los costes, costes que la empresa podría haberse ahorrado en caso de estar comunicada correctamente con el resto de los participantes de la cadena de suministro.

Aunque en un primer momento parezca que solo haya que comunicarse con el cliente, dado que es quien que va a comprar el producto y en el que se basarán las producciones, en realidad, la información debería fluir desde el principio, hasta el fin de la cadena de suministro, incluyendo también al proveedor. Esto se puede ver claramente si se piensa en la repercusión que tendría si el proveedor no suministrase el número correcto de piezas, debido al desconocimiento de la demanda del cliente. Si se tuvieran menos piezas de las deseadas, en primer lugar, esto podría suponer al fabricante una parada de línea, lo cual tiene un elevado coste, y además, perjudicaría la imagen de la empresa ya que los clientes no obtendrían lo demandado y su nivel de satisfacción descendería. Por lo tanto, todos los miembros de la cadena de suministro deben estar bien comunicados.

Gracias a que el proveedor y el fabricante están correctamente comunicados, el proveedor conoce la demanda a la que se enfrenta en cada situación, pues el fabricante la conocerá por su distribuidor, y éste por los clientes. Esto supone una gran ventaja tanto para el proveedor como para el fabricante ya que el proveedor siempre fabricará según lo que le exija el fabricante. Esto es muy beneficioso pues se minimizan los costes de inventariar para el proveedor y nunca habrá ni sobreproducciones ni infraproducciones, pues solo se producirá lo necesario.

Por otro lado, el fabricante también se beneficia de este flujo de información puesto que sabe que su proveedor siempre le va a suministrar todo lo que va a necesitar, con lo que no habrán retrasos a la hora de servir al cliente. Esto significa que se mantendrá la satisfacción del cliente y que éste seguirá confiando en la empresa a la hora de comprar de nuevo.

Añadiendo a esto último, el fabricante, como en este caso sería Ford, se ve muy beneficiado no sólo por lo mencionado anteriormente, sino porque al producir productos de gran tamaño y coste, los costes de inventariar son muy altos.

Siempre que estos costes se puedan minimizar al máximo, va a beneficiar a la empresa, sobre todo cuando ésta se encuentra en un sector tan competitivo como es el sector de la automoción y cualquier minimización en el coste que se pueda obtener, supone una gran ventaja para la empresa.

Como se ha podido ver, cualquier alteración o cualquier mala sincronización en la cadena de suministro pueden acabar en una catástrofe para la empresa, lo que siempre repercute en un aumento de los costes.

2.2.2. Estrategia de Planificación Agregada

Antes de empezar a producir en cualquier empresa, hay que decidir cuántas unidades y sobre qué productos se va a producir. Estas decisiones se deben tomar de forma premeditada y nunca aleatoriamente. Lo que se intentará será siempre gestionar la demanda, prediciendo los ciclos de consumo y adaptándose siempre a las necesidades del cliente. Esto suele ser bastante complicado puesto que aunque la empresa base sus predicciones en los resultados de la demanda de años anteriores, es posible que no se siga siempre el mismo patrón o que ocurran contratiempos que acaben perjudicando a la empresa, provocando una sobreproducción o en caso contrario, una infraproducción.

Añadido a lo anterior, existen determinados productos que tienen una demanda cíclica, es decir, que habrá momentos donde la demanda es mayor, otros en las que será menor, pero este ciclo es conocido y repetitivo. Con lo cual, es más fácil adaptarse a ese tipo de demandas. Este es el caso de los helados. Es evidente que a partir de Mayo y hasta Octubre, la demanda de helados sube rápidamente, teniendo su máximo auge durante los meses de Julio y Agosto, ya que son los meses más calurosos del año. En cambio, durante el resto del año, aunque hay gente que sigue tomando helados, éstos se consumen de manera mucho más reducida. Por ello, a la hora de predecir cuánta materia prima habría que comprar, cuántos operarios se necesitarían, etc. sería fácil hacerlo dado que aunque se desconozcan las cifras exactas, se conoce el ciclo que va a seguir la demanda.

Por otro lado, también existen otros productos, como en este caso son los automóviles, en los que la demanda es bastante inestable. En la siguiente gráfica se puede ver la evolución de las ventas de automóviles desde el año 1990 hasta el año 2013.

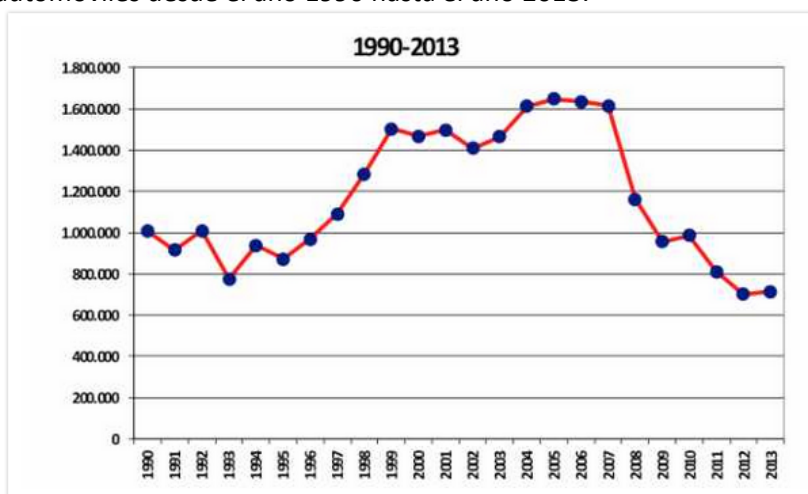


Gráfico 1. Evolución de las ventas de automóviles en España. (Fuente Suzuki.)

De esta gráfica se deduce que no hay ningún patrón ni tendencia, puesto que entre 1990 y 1995 la demanda era bastante estable y de ahí en adelante hasta 1999 aumentaba rápidamente. Más tarde, en el año 2000 hasta 2004 las ventas se volvían a estabilizar, sufriendo un pequeño aumento en 2004, aunque de nuevo volvían a estabilizarse. A partir de 2008 las ventas decayeron cada vez más, hasta llegar al mínimo en 2013, cuyas ventas son las peores registradas en los últimos 14 años.

Una vez estudiada la gráfica, se puede suponer que aunque se realicen previsiones, las demandas de un año para otro podrían variar drásticamente y con lo cual, la empresa no podrá adaptarse adecuadamente a la demanda. Por ello, hay que tener en cuenta cualquier factor exterior que pudiese afectar a la evolución de la demanda, y entonces prever qué sucederá. Por ejemplo, de esta gráfica se podría decir que el número de ventas de vehículos siempre ha tenido una tendencia en aumento hasta el 2008, donde de repente sufre una bajada contundente. Pero, analizando la situación desde un punto de vista global, en 2008 hay una causa que podría justificar este descenso. Esta causa es el estallido de la crisis económica mundial, y evidentemente eso afectó también al sector de la automoción.

Como el sector de la automoción se ve fuertemente afectado por la economía global, a la hora de prever la demanda, habrá que analizar cómo está la situación económica. Una vez se sepa si va mejorando o emporando, se podrá predecir una demanda u otra para la empresa. Hay que analizar bien esta situación puesto que no se quiere ni sobreproducir, ni todo lo contrario, ya que ambos conllevarían unos elevados costes. En el caso en el que hubiese sobreproducción, eso implicaría tener que inventariar más unidades. El coste de inventariar es excesivo en este sector debido al tamaño y coste que tienen los automóviles. Por otro lado, si se produce menos de lo demandado, no se servirá a tiempo al cliente y viendo que las ventas han decaído mucho en los últimos años, lo último que interesa es decepcionar al cliente y que éste realice menos compras en el futuro.

Añadiendo a lo anterior, hay que tener esto en cuenta a la hora de realizar la herramienta pues la demanda que se tendrá será simplemente una previsión de lo esperado y nunca se conocerá ésta a ciencia cierta.

2.2.3. Lean Manufacturing

El concepto básico de Lean es minimizar los desperdicios y optimizar los recursos. Se consideran como desperdicios: el transporte, inventario, movimiento, esperas, exceso de procesado, sobreproducción y defectos. Lean Manufacturing no es una técnica para reducir costes, sino más bien es una forma de pensar y actuar de toda la empresa, involucrándose toda ella, sin importar el rango de cada persona.

Hay que tener en cuenta también que Lean no solo está adaptado para la producción, si no que puede ser aplicado en cualquier empresa y proceso. Una empresa que tenga un enfoque Lean trabajará sólo con operaciones que aporten valor al cliente, y se deshará de aquellas que no lo hagan dado que es un coste del que no se obtiene beneficio alguno. Además, la empresa deberá centrarse en los procesos claves e incrementar su valor continuamente. Con lo cual, lo que Lean Manufacturing trata de hacer es quedarse únicamente con las operaciones con valor añadido y deshacerse del resto, mediante la colaboración de todos los miembros de la empresa.

Las técnicas de Lean Manufacturing se utilizan para dirigir la toma de decisiones por parte de la Dirección y se basan en aplicar los siguientes cinco principios:

1. Especificar qué es lo que supone Valor para el cliente. En primer lugar, no hay que pensar por los clientes, hay que averiguar qué es lo que éstos quieren. Hay que crear productos que identifiquen lo que el cliente considera como valor del producto ya que éste no pagará por algo que no considera valioso, aunque la empresa sí lo considere dado que ha invertido en ello. Por ello, todo lo que no se considere un valor para el cliente, será considerado como un desperdicio y algo en lo que no hay que centrarse, sino todo lo contrario, hay que deshacerse de él.
2. Identificar el mapa de la cadena de valor (VSM) para cada producto/servicio. El VSM (Value Stream Map) es una técnica gráfica que permite visualizar todo un proceso. El VSM detalla y entiende completamente el flujo, tanto de información, como de materiales necesarios, para que un producto o servicio llegue al cliente. Toda actividad que permita responder a una necesidad del cliente, representa un flujo de valor y las que no lo hagan, hay que eliminarlas.
3. Favorecer el flujo sin interrupción. Hay que lograr un movimiento continuo del producto/servicio a través de la corriente de valor y eliminar los obstáculos que hayan en el proceso. De esta manera se podrá reducir los tiempos de demora del flujo de valor. El objetivo es que el producto fluya al mismo ritmo en el que el cliente lo demanda.
4. Sistema PULL: Dejar que los clientes tiren de la producción. Se producirá en base a lo que el cliente demande y no al contrario. Esto supone unos costes de inventariar menores, pues solo se fabricará lo necesario y no habrá sobreproducciones.
5. Perseguir la perfección mediante la mejora continua. Hay que trabajar constantemente para conseguir unos tiempos de ciclo más cortos, obtener la producción exacta y con la mejor calidad posible, y centrar los esfuerzos en el valor para el cliente. Tiene que estar clara la idea de que siempre se puede mejorar. Además, hay que cumplir con las entregas de los productos, los requerimientos de los clientes y con la calidad deseada.

Entre otras, Lean ofrece las siguientes ventajas:

- Menos averías
- Menos accidentes
- Menos movimientos y traslados inútiles
- Menor tiempo para el cambio de herramientas
- Logra un mejor lugar de trabajo
- Mejora la productividad y la calidad del trabajo
- Menores niveles de inventario
- Mayor satisfacción por parte del personal dado que están más involucrados en todo el proceso
- Aumento de los beneficios
- Mayor satisfacción de los clientes

2.2.4. Milk-run

“Milk-run” es un método de suministro de diferentes productos provenientes de diversos proveedores. Es llamado “Milk-run” dado que hace referencia a años atrás, cuando diariamente en Estados Unidos había un lechero que visitaba los domicilios de toda la ciudad, recogiendo las botellas vacías que había en las puertas y sustituyéndolas por otras llenas.

En el “Milk-run” el lechero salía desde la fábrica con el camión lleno de botellas de leche y acudía a cada puerta. A pesar de que este trayecto era largo, una vez llegaba a la ciudad pasaba de puerta en puerta, sin necesidad de volver al almacén ni a cargar ni descargar puesto que al llegar a cada puerta, descargaba el material, y lo sustituía por el mismo número de botellas, pero en este caso llenas. Esto hace que el coste de transporte disminuya dado que la mayor distancia es la que hay desde la fábrica a los clientes con lo que si sólo se realiza una vez por día, este coste se verá minimizado.

El “Milk-run” que hacía el lechero antes, actualmente se aplica en el contexto de la logística. En la actualidad, en el sector de la automoción, la mayoría de fabricantes de coches tienen más de un proveedor y este método Milk-run consiste en que un solo camión pase por todos los proveedores y cargue con el material que cada proveedor produce. En cambio, el método que se utilizaba antes consistía en que cada proveedor individualmente enviaba un camión con su material producido directamente al fabricante de coches. Esto se puede ver en la siguiente imagen:

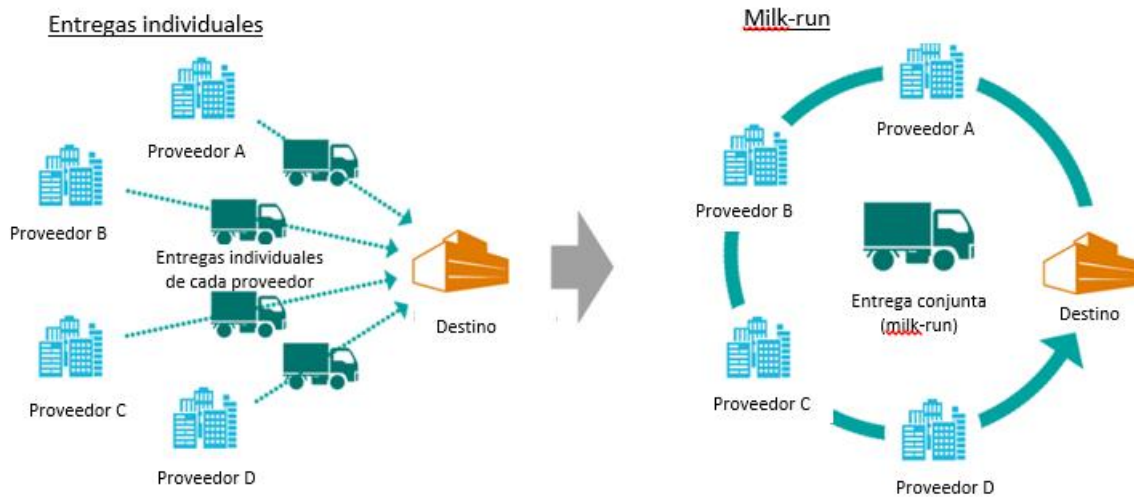


Imagen 4. A la izquierda se observa la manera en la que se suministraba al cliente anteriormente, mientras que a la derecha se muestra el suministro mediante el Milk-run.

En esta situación, en vez de utilizar cuatro camiones (uno por proveedor), se pasaría a utilizar sólo uno y a pesar de que éste haría una ruta más larga, siempre sería menor que la suma de las cuatro rutas de los otros camiones. Simplificando la situación, si cada proveedor tardase 1 día en llegar a la fábrica de destino, esto sería un total de 4 días. En cambio, al hacer el milk-run, el camión tardaría menos de 4 días ya que las distancias entre los proveedores no son tan grandes como entre el proveedor y la fábrica de destino.

Una vez entendido cómo funciona el “Milk-run”, se puede ver fácilmente que ofrece muchas ventajas. Algunas de esas ventajas son las siguientes:

1. Reducción de los costos de transporte debido al transporte consolidado, incluso si se transportan cantidades pequeñas.
2. Mayor precisión de la entrega de las mercancías Just-In-Time debido a la sincronización.

3. Mejora de la velocidad de carga del vehículo y reducción de la distancia total recorrida. Puede recoger el pedido de varios proveedores del fabricante, y al coordinarlos, mejora el abastecimiento, siendo éste más ágil y flexible, además de mejorar la capacidad de respuesta y la eficiencia del sistema del fabricante.
4. Reducción de los niveles de inventario en proceso, utilizando la logística de terceros, tanto por parte del proveedor, como por la del fabricante.

Esta idea del Milk-run también se utilizará en el desarrollo del proyecto para suministrar los materiales desde el almacén a los distintos puestos de trabajo.

Situación antes de aplicar Milk-run – entregas individuales:

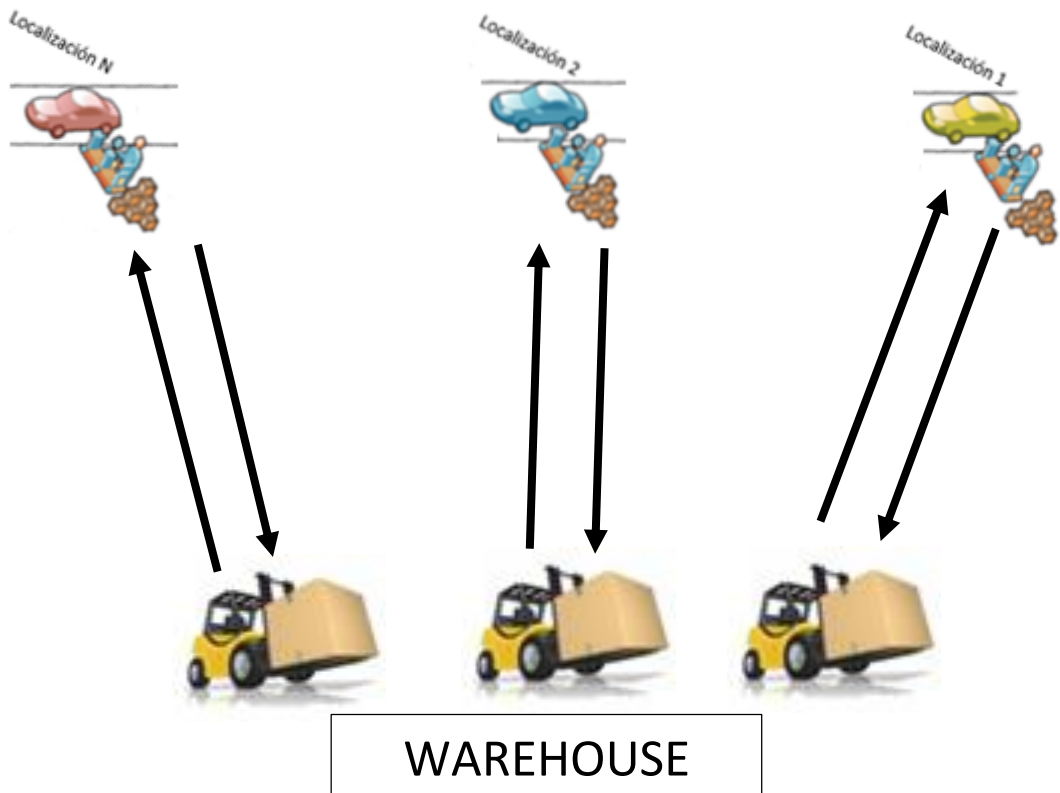


Imagen 5. Entregas individuales

Esta es la situación que habría si se realizasen las entregas individuales. En esta situación habría una carretilla para servir a cada localización. Se utilizaría una carretilla para atender la demanda de cada localización pues cada puesto de trabajo tiene unas necesidades distintas al resto de puestos, tanto en el número de componentes como en el tipo. Por lo tanto, el tener una carretilla para cada localización aseguraría el servir a tiempo la demanda de cada puesto.

La distancia conjunta recorrida por todas las carretillas al haber realizado una ruta individual desde el “warehouse” hasta el puesto de trabajo y regreso de nuevo, es muy superior a la distancia recorrida siguiendo una metodología “milk-run”, donde una sola carretilla partiría del “warehouse” e iría pasando y satisfaciendo la demanda de cada puesto de trabajo. Ésta sólo regresaría al “warehouse” para volver a cargar con material para suministrar los puestos que lo demanden.

Aplicando milk-run:

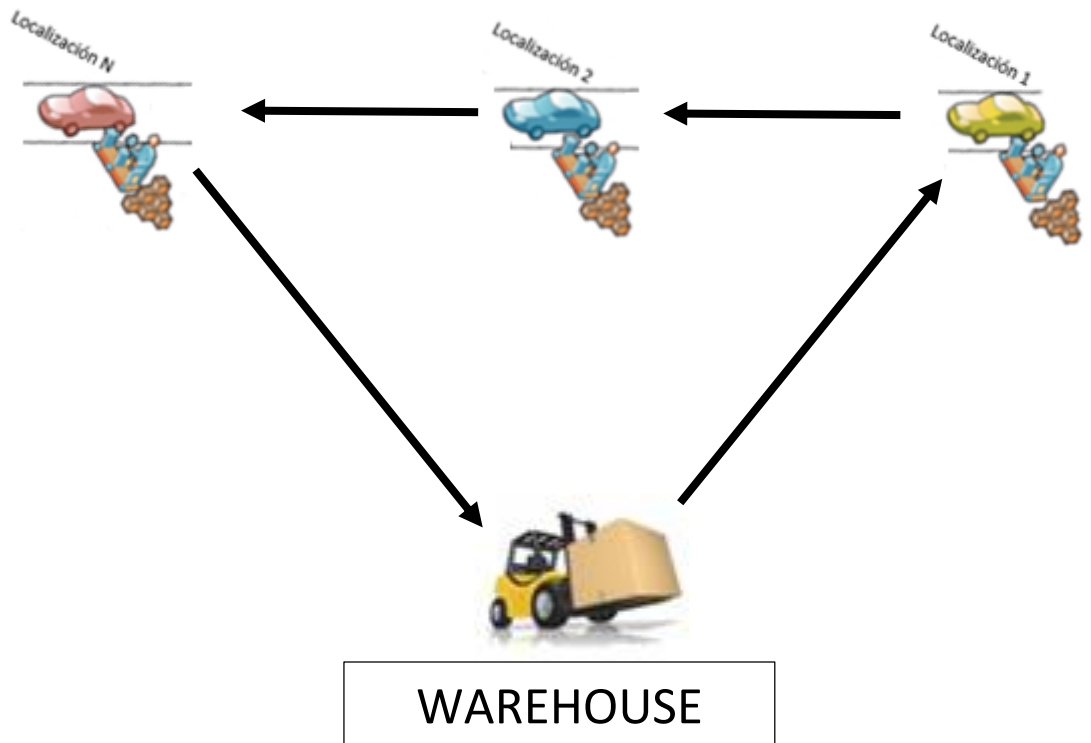


Imagen 6. Entregas por Milk-run.

Mediante lo explicado anteriormente y esta imagen se puede observar como el aplicar el “milk-run” reduce los costes. Estos se ven reducidos de dos maneras. En primer lugar, el pasar de utilizar una carretilla por cada a localización a utilizar varias por localización (intentando siempre reducir al máximo el número de éstas), resulta en una disminución de los costes por vehículo ya que se utilizarán menos carretillas y en consecuencia se necesitarán menos carretilleros. Además, como las distancias son menores entre las localizaciones y sólo se realizará un viaje al “warehouse”, el coste por el total de distancias recorridas también se verá reducido.

3. ANÁLISIS DE LA SITUACIÓN INICIAL

3.1 Análisis General del Proceso de Abastecimiento

Como se ha citado anteriormente, este trabajo está situado en la planta de montaje de Ford Almussafes y más concretamente en el área de Trim.

Ford Almussafes es una empresa de grandes dimensiones, que produce aproximadamente 1.600 automóviles por día. Teniendo en cuenta este volumen de producción, es lógico pensar que tiene un gran número de proveedores para poder suministrar todas las piezas que necesita. Ford cuenta con múltiples proveedores no sólo porque el coche esté compuesto por varios componentes, sino porque no se puede permitir el riesgo de tener sólo un proveedor por producto dado que si éste fallara, ya fuera porque no tiene material suficiente o por cualquier otro imprevisto, la fábrica tendría que parar, y esto sí que conllevaría un gran coste para la empresa.

Añadiendo a lo anterior, todos estos proveedores producen las cantidades exactas que Ford precisa y en el momento en el que lo exija, para así mantener la estrategia JIT durante toda la cadena de suministro y minimizar los costes de inventariar. Una vez Ford requiere de material, los proveedores cargan los camiones con los productos demandados y descargan en una misma área en la planta de Ford. Esta área será llamada en el trabajo como “Warehouse”. Una vez se encuentran los productos en el warehouse, a medida que son solicitados por las diferentes líneas y puestos de trabajo, son transportados y depositados en el puesto de trabajo correspondiente.

3.2 Análisis de los Dispositivos de Mantenimiento (carretillas)

Las carretillas más utilizadas en esta área son eléctricas y de bajo nivel sonoro, pues al trabajar en naves prácticamente cerradas y en proximidad a los operarios de la línea de montaje, la ausencia de gases de escape y la reducción del nivel sonoro son una prioridad. Otro requisito básico que tienen es que cuentan con robustas baterías de gran potencia con alta capacidad acumuladora y larga vida que proporcionen suficiente energía para un turno de 8 horas. Los modelos de carretillas eléctricas utilizados son la carretilla recogepedido y la carretilla elevadora retráctil.



Imagen 7. Carretilla recogepedido.
(Imagen extraída Jungheinrich)

Las carretillas recogepedido no limitan sus prestaciones al recogepedido sino que sirven también para otras tareas como el transporte horizontal y el depósito y la recogida de palets.

La carretilla retráctil es idónea para centros de distribución, almacenes con pasillos estrechos, o para la elevación de cargas hasta grandes alturas.



Imagen 8. Carretilla elevadora retráctil.
(Interempresas media, s.l.)



Imagen 9. Transpaleta manual.(Yale)

En esta área también se cuenta con algunas transpaletas manuales como medio de apoyo de picking. Diseñadas para manejarse con facilidad, permiten el movimiento seguro y eficiente de las mercancías con muy poco esfuerzo.

En la siguiente tabla se muestran las diferentes características de los medios de manutención:

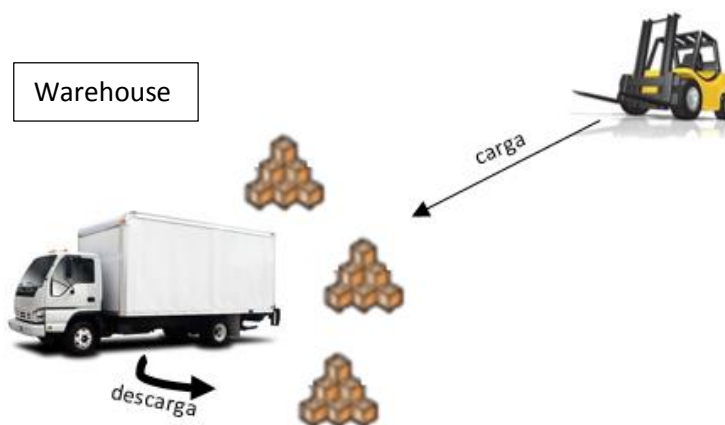
	Velocidad máxima	Capacidad máxima	Altura de elevación máxima
Carretilla recogepedidos	11 km/h	2,0 t	10,50 m
Carretilla elevadora retráctil	14 km/h	2,5 t	12,75 m
Transpaleta manual	En función del operario	1,0 t	0,80 m

3.3 Descripción Detallada del Proceso de Abastecimiento

En el área de TRIM, existen diversos puestos de trabajo dónde en cada uno de ellos se realizan unas funciones distintas y por eso, cada puesto necesitará aprovisionarse de unas piezas diferentes y seguramente en distintos momentos del tiempo. Esto hace que el proceso de suministrar las piezas en el momento adecuado sea tan complejo.

Además, como el plan de producción puede variar de un día a otro, dado que éste depende de la demanda de los clientes en esos momentos, la programación de suministro cambiará continuamente y cada puesto de trabajo necesitará aprovisionarse de distinta manera cada día. Esto hace que la situación se complique considerablemente y que sin una herramienta como la que se va a desarrollar en este TFG, resulte muy complicado planificar adecuadamente el aprovisionamiento de cada puesto de trabajo, intentando simultáneamente reducir los propios costes de la tarea de suministro.

La situación planteada se puede ver en las siguientes imágenes:



I

En la imagen superior se puede ver a un camión descargando. Este camión representa los distintos camiones provenientes de los proveedores descargando el material necesario para montar el automóvil. Como se ha mencionado anteriormente, estas piezas son entregadas y descargadas en el “warehouse”. Una vez descargadas y requeridas en línea, la carretilla vacía llega a la zona para aprovisionarse del material necesario.

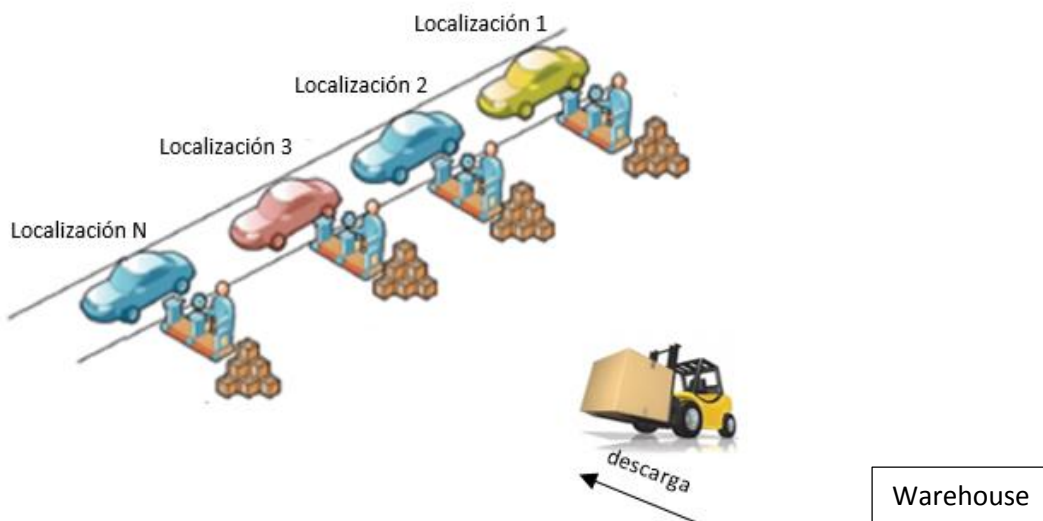


Imagen 11 . Una vez la carretilla elevadora ha cargado el material, ésta se dispone a dejarlo en la línea correspondiente.

En esta segunda imagen, la carretilla ya cargada con el material que necesita cada puesto de trabajo, sale desde el “warehouse” hasta la estación de trabajo correspondiente. Esta reposición sólo se realizará cuando el carretillero detecte (ya sea mediante las tarjetas identificativas de cada material, visualmente o por el botón de alarma) que no hay suficiente stock del material.

En el área Trim de la planta de montaje de Ford Almussafes, para garantizar el abastecimiento de las diversas piezas de mayor tamaño requeridas en cada puesto de trabajo se sigue el sistema de stocks mínimos, mientras que en las piezas de menor tamaño se sigue el sistema de reposición.

A continuación, se describen las tres formas distintas de transmitir las necesidades de cada puesto en la línea de montaje, dependiendo del tamaño de las piezas y de la frecuencia en que se consuman las mismas.

1. Piezas de tamaño reducido que se encuentran en estanterías. Un ejemplo de éstas sería un tornillo. Estos materiales son de media/alta rotación y para que siempre hayan, se sigue el siguiente proceso: cuando un operario se queda sin material, éste accede a la estantería a coger la siguiente pieza o el siguiente conjunto de materiales. Todo conjunto de materiales contiene con una tarjeta identificativa. Una vez el operario coge el material del que quiere proveerse, retira la tarjeta y la deja a la vista para que cuando pase el carretillero por esa estantería, sepa de qué material debe reponer las existencias y en qué cantidades.

2. Piezas opcionales. Este tipo de piezas son las que se utilizan sólo para algunos modelos de la línea. Como son de baja rotación (son de bajo consumo), el carretillero no está pendiente de su reposición. Por eso, cada pieza cuenta con un mínimo establecido y si el número de piezas baja de ese mínimo, el operario pulsa un botón alertando al carretillero de ello. Este tipo de aviso es muy eficiente para este tipo de piezas, pues el carretillero puede estar realizando otras operaciones sin preocuparse de su reposición y el operario nunca se quedará sin suministrar a tiempo.

3. Piezas de alta rotación y tamaño estándar. Este tipo de piezas se consume continuamente y está incluido en cualquier modelo. Son piezas no tan pequeñas como los tornillos, pero tampoco tan grandes como un asiento. Un ejemplo de estas piezas sería los pilares. Todos los coches llevan pilares para ajustar los techos, y además, se necesitan cuatro pilares por techo. La reposición de este tipo de piezas es visual. Los materiales vienen en carros y cuando el carretillero observa que quedan pocas piezas, repone el carrito vacío por uno lleno del “warehouse”.

4. PROPUESTA DE UN MODELO MATEMÁTICO PARA SU RESOLUCIÓN

4.1. Conceptos de la Modelización Matemática

4.1.1 Modelización Matemática

Como definición de la modelización matemática, se ha tomado la de Lesh, Harel (2003):
“Los modelos son sistemas conceptuales que generalmente tienden a expresarse utilizando una gran variedad de medios que interaccionan, los cuales pueden incluir símbolos, el habla, gráficos basados en ordenadores, diagramas o gráfico en papel. Su propósito es el de construir, describir o explicar otros sistemas. Los modelos incluyen:

- a) *Un sistema conceptual para describir o explicar los objetos matemáticos relevantes, las relaciones, acciones, patrones y regularidad que se atribuyen a la situación del problema a resolver*
- b) *procedimientos para generar construcciones útiles, manipulaciones o predicciones para conseguir metas reconocidas*

Los modelos matemáticos se distinguen de otras categorías de modelos principalmente porque se centran en las características estructurales (en vez de, por ejemplo, las características físicas, biológicas o artísticas) de los sistemas que describen.” (Harel y Lesh, Harel 2003, p.159).

Una vez vista la definición anterior, se puede deducir que la modelización matemática es simplemente una herramienta que transforma la realidad en un lenguaje que pueda ser resuelto. Lo que trata de hacer es representar una situación real mediante las matemáticas.

En cualquier empresa, sobre todo en el sector del automóvil dónde hay que ser competitivo y tratar de minimizar al máximo los costes para obtener la mayor rentabilidad posible, el poder utilizar modelos matemáticos que simulen la realidad, es muy beneficioso para ellos. Esto es debido a que estas empresas se suelen encontrar en situaciones muy distintas cada día. Estos diferentes cambios de situación que sufren continuamente provocan no saber cómo actuar en cada momento y que por lo tanto, sea complicado saber cómo enfrentarse a ellos correctamente. Hay que tener cuidado siempre ya que un error en la toma de decisiones podría significar un gran coste para la empresa, y eso es lo último que ésta busca.

Por ello, durante las últimas décadas se ha visto un incremento en la utilización de modelos matemáticos para optimizar problemas. Los modelos matemáticos dan la oportunidad de simular la situación y anticiparse a qué podría ocurrir. Esto resulta muy beneficioso puesto que mediante la solución que se obtenga se decidirá actuar de una manera u otra. Los modelos matemáticos resultan tan útiles que tanto en Europa como en Norte América, el uso de éstos para la planificación de la distribución produce una gran reducción de los costes de transporte (de un 5-20% menor coste).

Añadiendo a lo anterior, todo esto es posible desde hace tan sólo unas décadas atrás, debido a que anteriormente ni existían ordenadores, o si lo hacían, no eran de las características que tienen ahora puesto que ahora vienen con hardware y softwares más potentes.

Los modelos matemáticos pueden clasificarse de diversas maneras y las principales maneras de hacerlo son las siguientes:

- Cuantitativo o cualitativo: un modelo cuantitativo es aquel que utiliza números para representar la situación, utilizando entre otros, fórmulas y algoritmos. Este será el caso

de la mayoría de los modelos. En cambio, un modelo cualitativo se caracteriza por utilizar símbolos que representan cualidades no numéricas.

- Lineales o no lineales: cuando todos los operadores son lineales, el modelo será lineal. En el momento en el que al menos uno de ellos sea no-lineal, el modelo dejará de ser lineal. Siempre se intentará que los operadores sean lineales, ya que, por lo general, es bastante más simple plantear y resolver un modelo lineal, que uno no lineal.
- Estocásticos o deterministas: un modelo determinista siempre está definido por unos parámetros y variables de entrada fijas, por lo que las variables de salida también lo serán y con lo cual, el modelo siempre se comportará de la misma manera. En cambio, un modelo estocástico se define mediante variables de estado representadas por distribuciones de probabilidad, y por lo tanto, cada vez que se ejecute el modelo, éste se comportará de una manera distinta, teniendo unas variables de salida modificándose en cada ejecución.
- Estáticos o dinámicos: un modelo estático, como su nombre bien indica, no varía con el tiempo, mientras que el dinámico sí. Para definir las variaciones en el tiempo de los modelos dinámicos, se suelen utilizar ecuaciones diferenciales.

Una vez se sabe cómo va a ser el modelo, es decir, si va a ser cuantitativo o cualitativo, estático o dinámico, y antes de la creación de uno, hay que tener en cuenta los diferentes elementos de los que éste está formado. Estos son:

- Variables de decisión: son las incógnitas que debe resolver el modelo. Hay dos tipos de variables:
 - o Variables de entrada: estas variables son conocidas por el usuario y son los valores que se le da al sistema para alimentarlo.
 - o Variables de salida: estas variables son las incógnitas que el modelo trata de resolver. Una vez el modelo sea ejecutado y resuelto, en la solución que muestra estará el valor de estas variables. Las variables podrán tomar distintos valores dentro de un rango, dependiendo siempre de si se han definido como enteras o como binarias. Las variables enteras pueden tomar como valores: números enteros, en los que si el valor de la variable es distinto a cero, será una variable básica. En cambio, si la variable=0, será una variable no básica. En cuanto a las binarias, como su nombre indica sólo pueden tomar dos valores: 0 o 1.
- Constantes: existen 4 tipos de constantes: aritméticas, conjunto de números, reales y de caracteres.
- Restricciones: son las relaciones que existen entre las variables de decisión y los recursos disponibles. Éstos dan sentido a la solución del problema y acotan las variables de decisión a valores factibles. Esto permite al modelo dar alguna solución ya que si no existieran las restricciones, la función objetivo siempre daría el mejor resultado posible, es decir, al minimizar diría que el coste es 0 y al maximizar, los beneficios serían lo mayor posible.
- Parámetros: son valores que ya se conocen o que se pueden controlar y se utilizan para definir tanto las variables de decisión, como las restricciones.
- Función objetivo: este es el objetivo del problema, lo que realmente se está buscando. En la mayoría de casos, se suele querer minimizar o maximizar (como sería a la hora de realizar un proyecto, donde se querría o maximizar beneficios, o minimizar coste) aunque siempre se puede optimizar la función.

Además de todo lo mencionado anteriormente, cuando se cuente con un problema lineal y cuantitativo, hay que añadir la restricción de no negatividad. Esta restricción hará que el sistema no pueda coger ningún valor menor a cero, lo cual es práctico puesto que no tendría ningún sentido que si se estuviera intentado calcular, por ejemplo, el número de camiones necesarios para transportar una mercancía y el resultado fuera -3, esto carecería totalmente de sentido alguno y habría que volver a definir el modelo para ajustarlo.

Por último, una vez ya es conocido cómo va a ser el modelo y cómo se formará, se puede proceder a la creación del mismo. Para hacerlo, hay que tener en cuenta los siguientes pasos:

1. **Plantear la situación inicial:** se debe empezar a plantear la situación teniendo en cuenta qué información es estrictamente necesaria y de cuál hay que deshacerse puesto que no aporta nada. Además, en esta fase hay que saber qué es lo que se quiere obtener de este modelo. Lo que se desea obtener es siempre la función objetivo. Aunque esta fase parezca simple, hay que tener cuidado a la hora de definirla puesto que un error aquí significaría que todo el modelo es erróneo y no sería útil su resolución ya que se calcularía algo que en realidad no se deseaba calcular.
2. **Formulación del modelo:** toda la información con la que se cuenta hay que transformarla. Aquí se deben declarar todas las variables, parámetros y restricciones. También hay que incluir todas las suposiciones que se hagan.
3. **Resolución del problema:** una vez el modelo quede planteado, sólo queda resolverlo. Existen dos maneras para llevar a cabo su resolución. La primera, siempre que el modelo no sea demasiado complejo, se podrá resolver a mano. En cambio, cuando la complejidad del problema es mayor, habría que utilizar programas informáticos que mediante algoritmos podrían llevar a cabo su resolución. Existen varios programas que pueden utilizarse para su resolución, como por ejemplo, Mathematical Programming Language (MPL) o Algebraic Modeling Language for Mathematical Programming (AMPL).
4. **Interpretación del resultado:** cuando el modelo es resuelto, no sólo el valor de la función objetivo es devuelto, sino que el modelo devuelve también los valores de las variables de salida. Hay que analizar estas variables de salida y ver si el resultado es lo que se esperaba.
5. **Evaluación de la solución:** dependiendo de la interpretación que se haya deducido de los resultados, se evaluará la calidad del modelo; en otras palabras, se verá si el modelo es efectivo o si en cambio no es útil tal y como está actualmente definido.
6. **Redefinir el modelo:** una vez interpretados los resultados y evaluada la solución, se debe considerar la posibilidad de mejorar el modelo. En este paso hay que intentar encontrar la mejor manera de formular el problema, viendo si hay que añadir o prescindir de alguna restricción para utilizar el mejor modelo posible, y en consecuencia, obtener los mejores resultados posibles. Aunque se haya considerado que el modelo está bastante bien definido pues ha proporcionado los resultados esperados, hay que seguir buscando maneras de intentar mejorar el modelo.
7. **Evaluación de los resultados:** este es el paso final, con lo que hay que sacar las conclusiones finales. Aquí se analizará si el modelo resuelve la situación planteada al principio.

A continuación se muestra el diagrama con los pasos que hay que seguir a la hora de crear un modelo matemático:

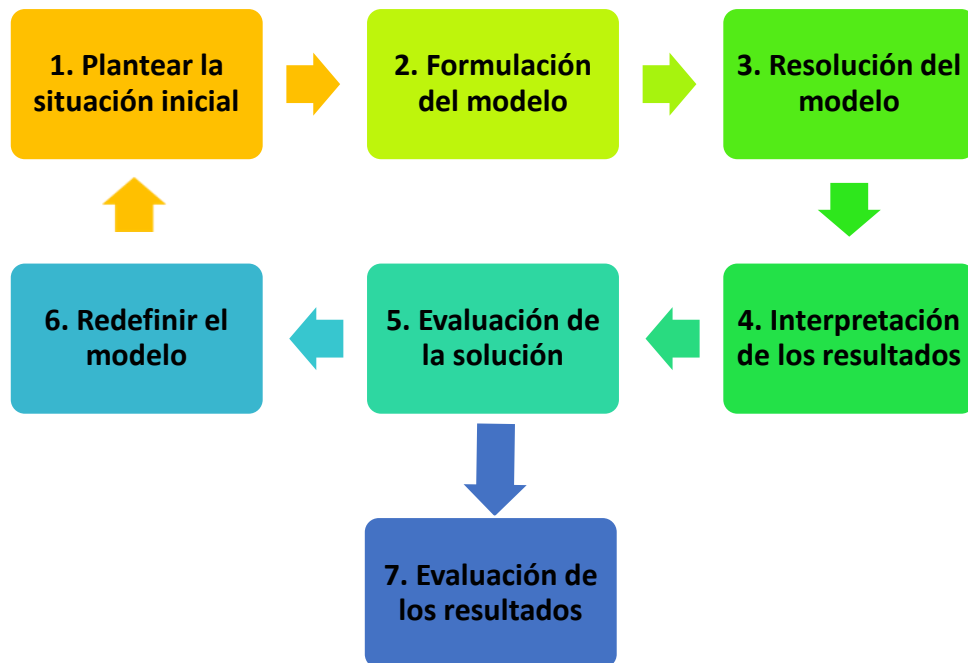


Diagrama 4. El ciclo de Berry, Davis (1996) representando los pasos a seguir al realizar un modelo matemático.

Una vez conocido el proceso de la creación de un modelo matemático y saber en qué situaciones se utiliza, se van a presentar una serie de ventajas que conlleva la utilización de éstos:

- El proceso de modelización puede ser una experiencia de aprendizaje, es decir, puede aportar una mayor comprensión y conocimiento de la situación dado que ésta se estudia con más detenimiento y además, desde otros puntos de vista.
- Permite simular una situación. Esto significa que se puede predecir la situación y si diera un resultado negativo, se sabe que hay que reconducir la situación y evitar acabar en ese punto.
- Se pueden evaluar varias alternativas ya que es capaz de evaluar diferentes soluciones en poco tiempo.
- A pesar de que se requiera tiempo e inversión para crearlos (software y hardware), esto es mucho menos costos que el fracasar o utilizar el método de ensayo y error.
- Permite estudiar situaciones complejas que en la vida real serían muy difíciles de probar. Por ejemplo, se podría modelar la situación de cuánto combustible se necesitaría para viajar a Marte. Esto sería imposible de realizar en la vida real puesto que se correría el riesgo de quedarse a mitad camino ya que se desconoce de la cantidad de combustible necesaria para llegar hasta Marte.
- Se puede utilizar en ingeniería y diseño de producto para investigar el efecto de los cambios sin producir físicamente un prototipo.

A pesar de que, como se ha visto anteriormente, los modelos matemáticos son muy útiles y sea beneficioso utilizarlos, también cuentan con algunas desventajas. Estas son las siguientes:

- Es imposible incluir todos los factores que podrían afectar a la situación, puesto que las posibilidades son infinitas.

- Hay que tener altos conocimientos para crear e interpretar modelos matemáticos complejos.
- En los modelos se asume que el futuro será como el pasado, es decir, que todo se comportará como se ha ido comportando hasta ahora y eso no siempre es cierto.
- En ocasiones, los datos de entrada pueden no conocerse con todo detalle, con lo que el resultado no será muy preciso y por lo tanto, poco útil.

4.1.2 Resolución Informática (MPL)

Como se ha mencionado anteriormente, una vez el modelo esté formulado, se puede proceder a su resolución. Para resolver los modelos existen dos posibilidades: resolverlo manualmente o realizarlo mediante herramientas diseñadas para resolver este tipo de problemas. Para problemas de pequeñas dimensiones, con una resolución manual sería más que suficiente. En cambio, para problemas de mayor envergadura, las herramientas son una mejor solución. Éstas facilitan el cálculo y reducen los tiempos de resolución, ya que utilizan varios algoritmos para obtener resultados de los modelos.

En este caso, como el problema es complejos, se ha decidido emplear el uso de una herramienta. Esta herramienta es la siguiente: Mathematical Programming Language (MPL).

MPL es un sistema de modelización que permite formular complejos modelos de una manera clara, concisa y eficiente. Una vez el modelo es introducido en el sistema, éste permite que se resuelva con los diferentes optimizadores comerciales disponibles en el mercado.

La elección del optimizador, también conocidos como “solver”, dependerá de qué se está buscando. Hay algunos que sólo sirven para modelos lineales, mientras que otros, sólo para los no lineales y por ello, hay que tener muy clara la clasificación del modelo. Añadiendo a esto, hay “solvers” que serán mejores que otros puesto que utilizan algoritmos y heurísticas más complejas para resolver los modelos. Aquí es donde viene bien contar con varios “solvers”, puesto que así se puede analizar cuál se adapta mejor al modelo creado y así encontrar la mejor solución posible.

Los diferentes “solvers” con los que cuenta MPL son:

- CPLEX 300: se utiliza para resolver modelos lineales. Hoy en día es uno de los optimizadores más populares del mercado. Ofrece soluciones completas que cuentan con prácticamente cualquier solución que se quiera obtener del modelo.
- Gurobi: potente optimizador que permite resolver modelos lineales y de entera mixta y explotar simultáneamente cualquier número de procesadores y múltiples núcleos.
- CoinMP: es un optimizador lineal y de números enteros. Tiene integrado un algoritmo pre-resuelto que elimina las restricciones y variables redundantes, mientras que por otro lado ajusta el resto de restricciones. El algoritmo que se utiliza para la resolución, emplea un número de cortes en las técnicas de factorización de Cholesky, lo que es especialmente útil para grandes problemas.
- SULUM: se utiliza tanto para resolver modelos lineales como de entera mixta. Para la resolución de modelos lineales, éste se basa en algoritmos, mientras que para entera mixta utiliza métodos más complejos como el branch and bound.
- LPSolve: es un optimizador lineal y de entera mixta. Se suele utilizar a la hora de resolver modelos simples, aunque permite resolver LP's de al menos 100.000 variables. Todavía no es considerado como un potente optimizador industrial, por lo que su uso se

restringe a la resolución de modelos pequeños, simples y cuando existen requerimientos presupuestarios.

- Ipopt: solver para problemas a gran escala no lineales en sistemas continuos. Ipopt resuelve los modelos mediante algoritmos de búsqueda local.
- Conopt: es un solver para problemas a gran escala no lineales y es uno de los pocos que se encuentra con estas características en el mercado. Normalmente, se utiliza para resolver modelos de 10.000 restricciones, pero es capaz de resolver modelos hasta con 1.000.000 de restricciones.
- Knitro: está diseñado para encontrar soluciones locales de problemas continuos y no lineales. A la hora de resolver, utiliza simultáneamente varios algoritmos, branch-and-bounds y detecta rápidamente cuándo un modelo no tiene solución.
- LGO: resuelve modelos no lineales bajo condiciones muy generales. Integra eficientes algoritmos tanto globales como locales.

MPL se ve beneficiado a la hora de ser escogido como sistema optimizador ante sus competidores puesto que al utilizar el mismo lenguaje que Windows, MPL y Windows son compatibles. Esta compatibilidad permite a MPL acceder a archivos tales como hojas de cálculo y bases de datos tanto para acceder a información que será utilizada en la ejecución del programa, como para exportar las soluciones a éstos.

4.2. Planteamiento de la Situación Inicial

En esta sección se expondrá el objetivo del modelo matemático y las restricciones a las que está sujeto.

A la hora de realizar el modelo, se ha intentado plasmar la situación inicial de la manera más real posible. Es evidente que no se puede tener definido el modelo de la misma manera que en la vida real, puesto que existen infinitas variables que podrían afectar a la situación. Un ejemplo de una variable que no se ha tenido en cuenta es que ocurren paradas inevitables de las máquinas. Como se ha mencionado anteriormente, esta es una de las desventajas que presenta la modelización matemática. Pese a ello, en el modelo se ha intentado incluir las variables que más afectarían al problema para que éste sea lo más preciso posible y se puedan obtener soluciones más realistas.

El modelo partirá de la situación inicial (planteada en la sección 3) y ésta es la siguiente:



Imagen 12. Una vez la carretilla elevadora ha cargado el material, ésta se dispone a dejarlo en la línea correspondiente.

Este modelo matemático se encargará de calcular los costes que supone el suministro por parte de los carretilleros a todas las localizaciones. El suministrar a cada localización tiene varios costes diferentes. El primero es por el número de carretillas que se utilizan. A este coste le siguen el de inventariar, el de desplazarse y el de transportar una unidad u otra. Todos estos costes se querrán minimizar.

Además de querer saber cuál sería el coste total de realizar este suministro, también se desea saber el número de carretillas que se deberían utilizar para minimizar los costes, y las rutas que deberían seguir éstos. Para conseguirlo, mediante la experimentación se probarán diferentes valores para el número de carretilleros, viendo si es más eficiente el tener más carretillero de los que se tiene en ese momento. El planteamiento sería el siguiente: contando con el número actual de carretillas, éstos harán más de una vuelta. Esto puede provocar retrasos, stock acumulado y un mayor coste debido al número de viajes que realizan. En cambio, si se dispusiera de más carretillas, éstas podrían depositar el material en el momento preciso, llegarían con menos retraso y realizarían menos viajes, aunque por otro lado habría que pagar el uso de una carretilla más y tener a un carretillero. Se considerará que todas las carretillas son idénticas, es decir, que todas pueden transportar la misma cantidad de material y que todas trabajan a la misma velocidad.

Para minimizar los costes, se planteará el suministro de la siguiente manera:

1. No se suministrará antes de tiempo. Hay que saber el momento exacto en el que el operario va a necesitar ser aprovisionado, y justo entonces suministrarle con el material requerido. No se realizará antes pues esto conlleva dos problemas. El primero es que como el operario tiene un determinado espacio para almacenar las piezas, si se suministrase antes de tiempo, llegaría un punto en el que las piezas no tendrían sitio para almacenarse y el operario acabaría dejando las piezas en lugares inadecuados, donde otro operario podría coger el material por equivocación, o la pieza podría dañarse ya que no está en su lugar adecuado y alguien podría tropezar con ella. Por otro lado, el suministrar antes de tiempo también repercute en un aumento de los costes de inventariar. Esto es perjudicial para Ford ya que sigue una estrategia de bajos costes y esto supondría elevar los mismos.
2. No se suministrará más tarde. En caso de suministrar una pieza tarde significaría que el operario no podría seguir haciendo su trabajo y puede que tenga que parar la línea y además, al cliente no le llegará el vehículo en el momento en el que lo deseaba, con lo que su nivel de satisfacción se verá afectado.

A la hora de la realización del programa, ciertos puntos deben ser aclarados:

- Como el modelo se puede ejecutar tantas veces como se desee o precise, el stock inicial en cada situación será el stock con el que se encuentre en ese momento en cada puesto.
- el número de localizaciones es fijo, con lo cual no se podrá modificar su número; no obstante, se harán simulaciones mostrando qué ocurre al aumentar el número de localizaciones.
- el número de componentes también es fijo.

Como se ha querido realizar una simulación lo más real posible, se respetará la capacidad del área Trim y por lo tanto se trabajará bajo capacidad finita.

El coste total del suministro se conocerá en la resolución del modelo mediante el valor de la función objetivo y las rutas que debería seguir cada carrito se podrán ver en el valor de las variables de salida.

4.3. Formulación del Modelo

Antes de mostrar el modelo, se han definido los parámetros, los índices y las variables para poder trabajar a partir de ellos. Estos son los siguientes:

Parámetros:

Parámetros	Definición	Valor
R	Número máximo de vehículos que pueden hacer una ruta	R=2
L	Localizaciones (puestos de trabajo)	L=4
J	Número máximo de componentes a servir en la línea	J=2
T	Número de periodos a considerar	T=10
M	Número muy grande	M=100.000

Tabla 2. Definición de los parámetros del modelo.

Índices:

Una vez se conocen ya cuáles son los parámetros y qué valores tienen, se pasa a definir los índices:

- $r=1\dots R$
- $l=1\dots L$
- $j=1\dots J$
- $t=0\dots T$

Variables:

Hay dos tipos de variables:

1. Variables enteras: estas son variables que almacenan valores enteros.
 - $f(j,l_1,l_2,r)$: es la cantidad de componente de tipo j transportado entre la primera y segunda localización (l_1 y l_2) en la ruta r ($\forall j, l_1, l_2, r / l_1 \neq l_2$)
 - $st(j,t,l_1)$: es la cantidad de componente de tipo j que hay en la primera localización (l_1) al principio del periodo t ($\forall j, l_1, r / l_1 \neq 0$)
 - $t(r,l_1)$: instante en el que la ruta r llega a la primera localización (l_1)
 - $c(j,l,r,t)$: cantidad de componente j suministrada a la localización l , por la ruta r en el periodo t .
2. Variables binarias: estas son variables que sólo pueden tener dos valores: 0 o 1.
 - $x(r,l_1,l_2) = 1$: la variable tomará el valor 1 si se va desde la primera localización a la segunda (l_1 a l_2) por la ruta r y tomará el valor 0 en caso contrario ($\forall r, l_1, l_2 / l_1 \neq l_2$)
 - $w(r,l_1) = 1$: la variable tomará el valor 1 si la ruta r pasa por la primera localización (l_1) y 0 en caso contrario ($\forall r, l_1$)
 - $a(r) = 1$: la variable tomará el valor 1 si se usa la ruta r y 0 en caso contrario ($\forall r$)

Además de estas variables, el modelo cuenta con unos datos de entrada:

Tiempos de desplazamiento:

$tdist(l1,l2)$: tiempo (medido en distancias) que se tarda en ir desde la primera localización (l1) hasta la siguiente (l2).

	0	1	2	3
0		6	7	8
1	6		2	1
2	7	2		3
3	8	1	3	

Tabla 3. Tiempos de desplazamiento.

Matriz de demanda:

$dem(t,j,l)$: es la demanda en el instante t del componente j en una localización l

Se supone que el tiempo de ciclo $TC=5$

	l=1		l=2		l=3	
	j=1	j=2	j=1	j=2	j=1	j=2
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	4	3	2	1	0	8
6	0	0	0	0	0	0
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	5	1	4	7	2	1

Tabla 4. Matriz de demanda.

Stock inicial:

$STO(l,j)$: stock inicial en cada localización l de cada componente j

	l=1		l=2		l=3	
	j=1	j=2	j=1	j=2	j=1	j=2
STO	2	3	5	3	1	4

Tabla 5. Stock inicial

Coste de almacenaje:

$HC(j)$: coste de almacenar cada componente j

	j=1	j=2
HC(j)	6	7

Tabla 6. Coste de almacenaje.

Otros datos de entrada:

- Coste por unidad de distancia: $C=10$
- Coste por unidad transportada: $MC=5$
- Coste de uso de vehículo: $A=20$
- Número de unidades máximas que se pueden transportar en un vehículo: $CAP=10$

Función Objetivo:

El objetivo de este modelo matemático es minimizar los costes en la planta de ensamblaje de Ford. En este caso, son cuatro los costes que se están minimizando. Estos son: los de las distancias recorridas, el transporte de los componentes, el coste del número de rutas y el de inventariar.

$$\min C_{total} = C_{distancia} + C_{transporte} + C_{rutas} + C_{inventariar}$$

Coste distancia:

El coste de la distancia se debe a que cuantos más metros se recorran, mayor tiempo se empleará en realizar esa acción. El hecho de emplear más tiempo significa tiempo que no se está utilizando en otras acciones y eso supone un coste para la empresa. Por todo esto, se quiere que las distancias sean las menores posibles.

$$\sum_{r=1}^R \sum_{l1=0}^L \sum_{l2=0}^L TC * Tdist(l1, l2) * x(r, l1, l2)$$

Coste transporte:

El coste de transporte se refiere a la suma de costes de transportar una unidad desde una localización a otra. Como existen varias piezas y cada una tiene un tamaño, eso significa que no tendrá el mismo coste el transportar una unidad de una pieza u otra.

$$MC * \sum_{j=1}^J \sum_{l1=0}^L \sum_{l2=0}^L \sum_{r=1}^R f(j, l1, l2, r)$$

Coste rutas:

Este coste es un coste lineal que aumentará a medida que haya más rutas, dado que se necesitarán más carretillas y personal para llevar a cabo la disposición del material.

$$A * \sum_{r=1}^R a(r)$$

Coste inventariar:

Este coste se encuentra siempre, sea cual sea la empresa, y se debe al hecho de tener en el almacén material que no está siendo utilizado. Esto significa que hay dinero que no está siendo utilizado y además, espacio utilizado que ya no se puede utilizar para otras cosas. Como se ha

visto anteriormente, como en este caso se está hablando de automóviles, el coste de tenerlos inventariados supone un gran coste ya que son productos grandes que ocupan mucho espacio y son costosas.

$$\sum_{j=1}^J \sum_{t=1}^T \sum_{l1=1}^L HC(j) * st(j, t, l)$$

Con lo cual, la Función Objetivo (F.O.) sería la siguiente:

$$\min = \sum_{r=1}^R \sum_{l1=0}^L \sum_{l2=0}^L TC * Tdist(l1, l2) * x(r, l1, l2) + MC * \sum_{j=1}^J \sum_{l1=0}^L \sum_{l2=0}^L \sum_{r=1}^R f(j, l1, l2, r) + A * \sum_{r=1}^R a(r) + \sum_{j=1}^J \sum_{t=1}^T \sum_{l1=1}^L HC(j) * st(j, t, l)$$

Restricciones:

Como se ha mencionado anteriormente, este problema está sujeto a varias restricciones. Éstas son las siguientes:

Sabiendo que la localización l1 y l2 no son la misma y que la localización l1=warehouse, siempre existirá al menos una ruta que pasará desde el warehouse hasta la localización l2:

$$\sum_{r=1}^R \sum_{l1=0/l1 \neq l2}^L x(r, l1, l2) = 1 \quad \forall l2 > 0 \quad (1)$$

Sabiendo que la localización l1 y l2 no son la misma y que la localización l2=warehouse, siempre existirá al menos una ruta que pasará desde la localización l1 hasta el warehouse. Mediante esta restricción y la anterior se trata de imponer que cualquier ruta que exista deberá empezar y acabar en el warehouse:

$$\sum_{r=1}^R \sum_{l2=0/l2 \neq l1}^L x(r, l1, l2) = 1 \quad \forall l1 > 0 \quad (2)$$

Al menos existirá una ruta que sirve cada localización:

$$\sum_{r=1}^R w(r, l1) = 1 \quad \forall l1 > 0 \quad (3)$$

Si una localización es servida por una ruta, a excepción del warehouse, todas deben tener un sucesor:

$$w(r, l1) = \sum_{l2=0/l2 \neq l1}^L x(r, l1, l2) \quad \forall r, l2 > 0 \quad (4)$$

Sabiendo que la localización de la que se parte a la que se llega no son iguales, se deberá cumplir que si una ruta llega a una localización, ésta tendrá que después salir desde ahí para poder ir a otra localización:

$$\sum_{l1=0}^L x(r, l1, l2) = \sum_{l1=0}^L x(r, l2, l1) \quad \forall r, l2 \neq l1 \quad (5)$$

Existen tantas rutas que van al warehouse como número de vehículos:

$$\sum_{r=1}^R \sum_{l1=1}^L x(r, l1, 0) = \sum_{r=1}^R a(r) \quad (6)$$

Existen tantas rutas que salen del warehouse como número de vehículos:

$$\sum_{r=1}^R \sum_{l2=1}^L x(r, 0, l2) = \sum_{r=1}^R a(r) \quad (7)$$

Si el vehículo visita al menos una localización, existirá la ruta “r”:

$$\sum_{l1=0}^L \sum_{l2=\substack{0 \\ l2 \neq l1}}^L x(r, l1, l2) \leq M * a(r) \quad \forall r \quad (8)$$

Las rutas se numeran en orden creciente, es decir, la primera ruta que se cree será r1, la siguiente r2 y así:

$$a(r) \geq a(r + 1) \quad \forall r \quad (9)$$

Debe haber como mucho tantas rutas como número de vehículos:

$$\sum_{r=1}^R a(r) \leq R \quad (10)$$

Calcula el tiempo que tarda en llegar la ruta (r) desde el WH(l=0) a la primera localización l2:

$$t(r, l2) \geq Tdist(0, l2) - M * (1 - x(r, 0, l2)) - M * (2 - w(r, 0) - w(r, l2)) \quad \forall r, l2 \neq 0 \quad (11)$$

Calcula el tiempo que tarda en llegar la ruta r desde la primera localización ($l1$) a la segunda localización ($l2$):

$$t(r, l2) \geq t(r, l1) + Tdist(l1, l2) - M * (1 - x(r, l1, l2)) - M * (2 - w(r, l2) - w(r, l2)) \quad \forall r, l1, l2 / l2 \neq l1 \quad (12)$$

Establece la capacidad máxima que pueden transportar los carritos por las rutas debido a la capacidad que tiene cada carrito es limitada:

$$\sum_{j=1}^J \sum_{l2=1}^L f(j, 0, l2, r) = CAP \quad \forall r \quad (13)$$

La cantidad de componente j que se transporta entre dos localizaciones ($l1$ y $l2$) más la que se suministra en la primera localización ($l1$), debe ser menor que la que se transportó entre l y $l1$, siendo siempre localizaciones diferentes. Esto representa la cantidad de componentes “ j ” que se dejan en cada estación:

$$f(j, l1, l2, r) + c(j, l1, r) \leq f(j, l, l1, r) - M * (1 - x(j, l, l1)) - M * (1 - x(j, l1, l2)) - M * (3 - w(r, l) - w(r, l1) - w(r, l2)) \quad \forall j, l, l1, l2, r / l \neq l1 \neq l2 \quad (14)$$

Define el stock en el primer instante del tiempo ($t=1$); éste será el stock inicial, menos la demanda, mas la cantidad que se ha suministrado a esa primera localización $l1$:

$$st(j, 1, l1) = ST0(j, l1) - dem(1, j, l1) + \sum_{r=1}^R c(j, l1, r, 1) \quad \forall j, l1 \quad (15)$$

El stock que queda en cada localización $l1$ de cada componente j a final del periodo t :

$$st(j, t, l1) = st(j, t - 1, l1) - dem(t, j, l1) + \sum_{r=1}^R c(j, l1, r, t) \quad \forall j, l1, t > 1 \quad (16)$$

El stock que hay de cada componente en cada localización ($l1$), debe ser mayor que la demanda de ese componente (j) en esa localización en cada periodo (t):

$$st(j, t, l1) \geq dem(t, j, l1) \quad \forall j, l1, t > 1 \quad (17)$$

4.4 Resolución del Modelo

En primer lugar, se probarán dos optimizadores distintos para ver qué resultados proporcionan y si éstos devuelven resultados semejantes.

Resolución mediante CoinMP:

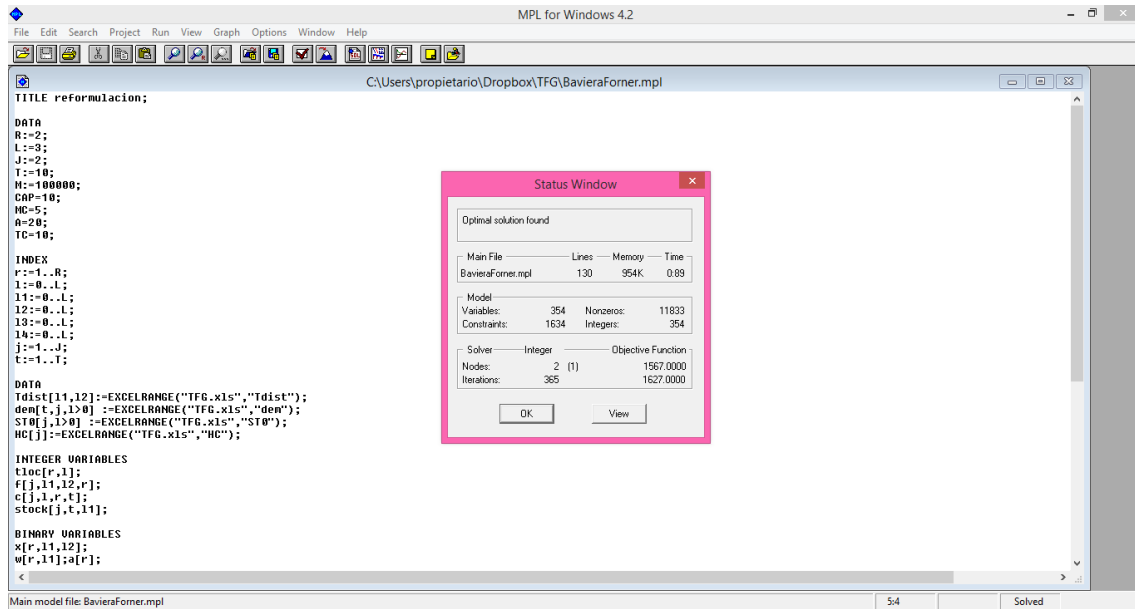


Imagen 13. Resolución mediante CoinMP.

Resolución LPSolve:

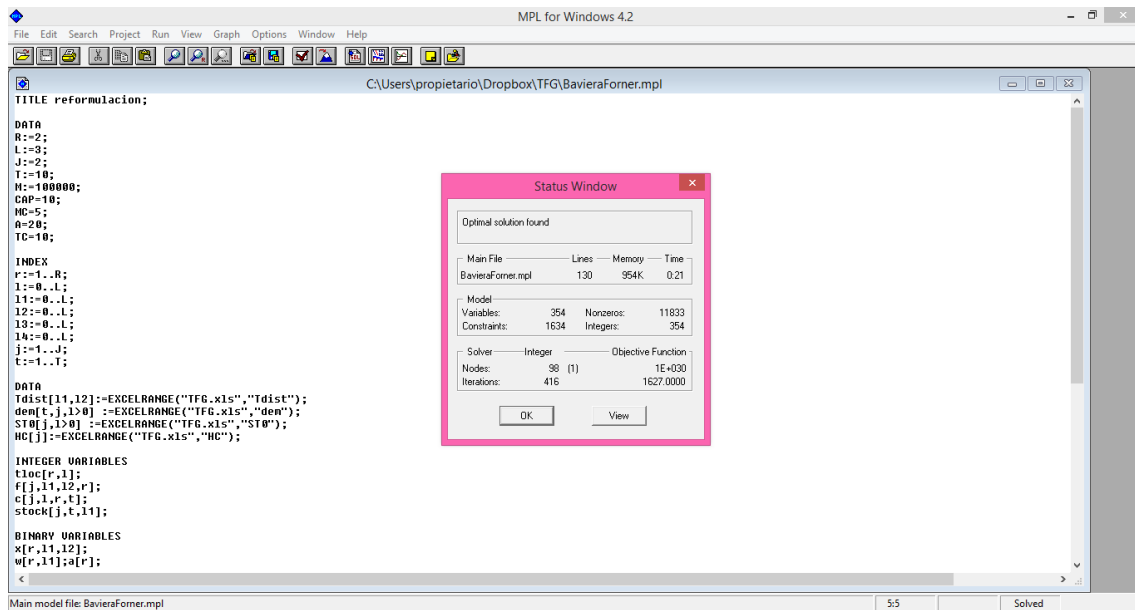


Imagen 14. Resolución mediante LPSolve.

Como se puede ver, utilizando dos optimizadores diferentes, ambos llegan a la misma conclusión. Los tres han encontrado una solución óptima y ésta es: 1651.00 €. Este será el mínimo coste que supondría abastecer tres localizaciones, con dos componentes distintos y con dos carretillas elevadoras.

Añadiendo a lo anterior, mediante la resolución del problema con los dos optimizadores se puede deducir que CoinMP es mejor a LPSolve ya que provee la misma respuesta en un tiempo menor. Por ello, será éste el que se utilizará como optimizador a partir de ahora.

A la hora de resolver, MPL devuelve no sólo el valor de la función objetivo, sino también el valor de las variables de salida. Dependiendo del tipo de variables que sean, éstas obtendrán un valor u otro. En el caso en el que fueran variables enteras, éstas podrían tomar cualquier valor mayor a cero, en cambio, si las variables fueran binarias, sólo podrían tener como valores uno o cero.

4.5 Interpretación de los Resultados

Aquí hay un ejemplo de ambas:

- Variable entera: $c(j,l,r,t)$: cantidad de componente j suministrada a la localización l , por la ruta r en el periodo t .

j	l	r	t	Activity	Reduced Cost
1	1	1	5	6	0
1	1	1	6	0	0
1	2	1	5	3	0
1	2	1	6	0	0

Tabla 7. Ejemplo con variable entera.

Mediante esta variable se muestra cuándo se suministra de un componente a una localización. Como el tiempo de ciclo es de cinco, sólo se suministrarán piezas cada cinco unidades de tiempo y es por ello por lo que en el momento $t=6$ no existe ninguna unidad suministrada.

- Variable binaria $w[r,l]$: variable binaria que tomará el valor de 1 siempre y cuando la ruta “ r ” pase por la localización “ l ” y tomará el valor de cero en caso contrario.

r	l	Activity	Reduced Cost
1	1	1	0
1	2	1	0
1	3	1	0
2	1	0	0

Tabla 8. Ejemplo con variable binaria.

De esta variable se puede deducir que la ruta 1 pasará por las localizaciones: 1, 2 y 3 y que en cambio la 2 no pasará por la 1.

De entre todas las variables que proporciona la resolución del modelo matemático, la variable que más interesa (además de la del coste total que supondría este suministro) es la que representa qué rutas se van a tomar. Esa variable es $x[r,l_1,l_2]$.

Variable $x[r,l1,l2]$:

r	l1	l2	Activity	Reduced Cost
1	0	1	1	0
1	0	3	1	0
1	1	0	1	0
1	2	0	1	0
1	3	2	1	0

Tabla 9. Resolución de las rutas (L=3, J=2, R=2).

Del resultado anterior se extraen las siguientes conclusiones:

- Existirán dos rutas.
- Todas las rutas empezarán y acabarán en cero (warehouse) ya que es una restricción que se le ha impuesto a todas las rutas.
- La ruta 1 irá de la localización cero a la uno y de vuelta a la cero y la ruta 2 partirá de la localización cero para ir a tres, de tres a dos y de dos a cero.
- Se pasa por todas las localizaciones.
- Se utilizará sólo una carretilla.

Carretilla 1. Ruta 1:

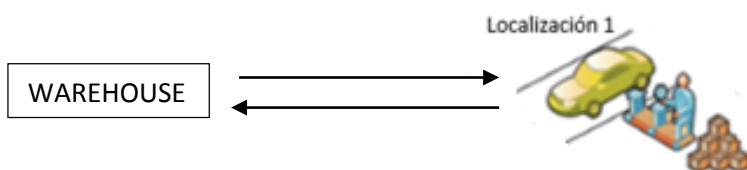


Imagen 15. Carretilla 1. Ruta 1 (L=3, J=2, R=2).

Carretilla 1. Ruta 2:

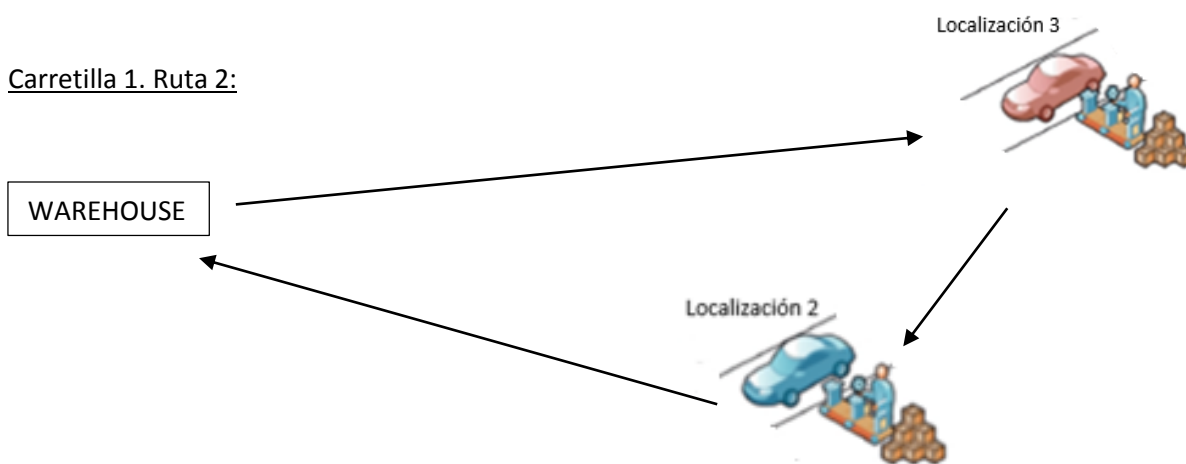


Imagen 16. Carretilla 1. Ruta 2 (L=3, J=2, R=2).

A continuación se muestran los resultados obtenidos al realizar varios experimentos:

1. $L=3, J=3, R=2$ – aumento del número de componentes

Coste: 1624,00€

Carretilla 1. Ruta 1: (0,1,0)

Carretilla 1. Ruta 2: (0,3,2,0)

Pese a que ahora se cuenta con un componente más, el modelo ha resuelto el modelo de la misma manera que si se tuviera dos componentes distintos, es decir, igual que en la resolución anterior.

2. $L=3, J=2; R=3$ – aumento del número de carretillas

Coste: 1677,00€

Carretilla 1. Ruta 1: (0,1,0)

Carretilla 1. Ruta 2: (0,2,3,0)

Esta solución era de esperar puesto que si ya con dos carretillas sólo utilizaba una, ahora también iba a mantenerse en sólo una aunque contara con la posibilidad de utilizar tres.

3. $L=4, J=2, R=2$ – aumento del número de localizaciones

Coste: 1771,00 €

Carretilla 1. Ruta 1: (0,4,3,0)

Carretilla 2. Ruta 1: (0,2,1,0)

Este aumento en el coste era de esperar pues ahora se va a suministrar a un puesto más. Esto implica mayores desplazamientos y mayor cantidad de piezas para inventariar.

4.6 Evaluación de los Resultados

Durante estas tres primeras experimentaciones, el modelo ha proporcionado una respuesta bastante rápida (tiempo de respuesta menor de un minuto), pero, en el momento en el que se ha aumentado el número de localizaciones a diez, el modelo ha tardado casi tres minutos, y además ha proporcionado una respuesta que carece de sentido. Esto se puede ver a continuación:

$L=10, J=2, R=2$ – aumento considerable en el número de localizaciones

Coste=3582,00€

Carretilla 1. Ruta 1: (0,5,7,0)

Carretilla 1. Ruta 2: (1,10,1)

Carretilla 1. Ruta 3: (2,9,2)

Carretilla 1. Ruta 4: (3,8,3)

Carretilla 1. Ruta 5: (4,6,4)

Este resultado carece de sentido pues como se ha mencionado con anterioridad, toda ruta realizada por una carretilla empezará y terminará en el “warehouse”. Mediante las rutas 2, 3, 4 y 5 se puede ver como esto no se cumple.

En conclusión, aunque la modelización matemática es una gran herramienta que permite resolver problemas complejos de manera óptima, también tiene ciertas limitaciones. En este caso, el modelo consigue encontrar una solución óptima para valores bajos de las variables, pero en el momento en que éstos aumentan el modelo ya no consigue resolver de manera óptima el problema planteado, por ejemplo, al aumentar las localizaciones a 10. Para poder resolver estas situaciones en que las variables toman valores más elevados, hay que buscar una solución alternativa, y por ello, se ha planteado una heurística que permite resolver este tipo de problema, por otro lado totalmente realístico. Esto se puede ver en el siguiente apartado.

5. PROPUESTA DE UNA HERRAMIENTA DE CÁLCULO DE RUTAS DE TRANSPORTE

5.1. Visual Basic for Applications (VBA)

5.1.1. Definición VBA

Visual Basic es un lenguaje de programación desarrollado por Alan Cooper. Las sintaxis que utiliza este lenguaje de programación proviene de “BASIC”, pero completada con códigos y comandos de otros lenguajes más modernos. Este lenguaje de programación tiene un apartado que está dedicado a la Programación Orientada a Objetos.

Añadiendo a lo anterior, Visual Basic for Applications (VBA) es el lenguaje de macros Visual Basic que se utiliza para programar aplicaciones Windows y que se puede encontrar siempre que se tenga Excel (Paquete de Office de Microsoft). Su utilidad principal es automatizar tareas cotidianas, así como crear aplicaciones y servicios de bases de datos para el escritorio.

Mediante las macros se saca más eficiencia a Excel. En muchas ocasiones, el Excel parece que no aporta todo lo que podría y de normal hay que hacer más trabajo del que se debería para resolver problemas. Mediante la programación en Visual Basic, muchos de estos problemas se pueden solucionar rápidamente y al final es la propia Excel la que trabajará para el usuario y disminuirá su carga de trabajo. El automatizar tareas repetitivas suele ser la razón más frecuente por la que el usuario utiliza VBA. Además, mediante VBA se pueden crear algoritmos para optimizar procesos (como va a ser el caso de este trabajo).

La utilización de este lenguaje es muy ventajosa por diversos motivos. El primero es que cualquier ordenador Windows cuenta con ella, con lo cual, no hace falta instalarla en cada dispositivo cada vez que se quiera ejecutar el programa creado. Esto es una ventaja enorme puesto que muchas herramientas de programación sólo funcionan en ciertos dispositivos o son de pago, mientras que el Visual Basic estará disponible siempre que se disponga de Microsoft Office Excel. Con lo cual, cualquier empresa tiene acceso a esta herramienta.

Además de esto, el lenguaje que utiliza Visual Basic es un lenguaje muy simple y puede ser utilizado tanto por expertos programadores, como por usuarios principiantes. Esto no sólo es positivo para cualquier usuario ya que está al alcance de todos, sino que además, al utilizar un lenguaje básico, si se deseara transformar a otro tipo de lenguaje más complejo, no llevaría mucho tiempo ni esfuerzo.

Otro punto positivo con el que cuenta VBA es que muchas empresas cuando llegan a tener un gran tamaño o utilizan grandes cantidades de información, deciden utilizar programas a medida. Estos programas a medida no siempre se adaptan como la empresa espera, con lo que no satisfacen sus necesidades; sin embargo, mediante VBA los programas desarrollados se adaptan sin problema a cada empresa, aportando versatilidad y flexibilidad.

5.1.2. Procedimiento inicial

Como se ha mencionado anteriormente, VBA se encuentra en Excel, y para empezar a programar sólo hay que seguir las siguientes sencillas explicaciones.

Hay que crear una macro, teniendo en cuenta con qué versión del Excel se está trabajando. Si se está operando con la última versión de Excel (2010), para ejecutar el Visual Basic sólo habrá que entrar en la pestaña “Programador” y seleccionar “Visual Basic”.



Imagen 17. Cómo crear una macro desde Microsoft Office Excel 2010

En cambio, si se está trabajando con cualquier versión más antigua, habría que entrar en “Controles” y después seleccionar “Botón de comando”. Una vez creado el botón, hacer doble click en él y ya nos conducirá a la pantalla principal de Visual Basic.

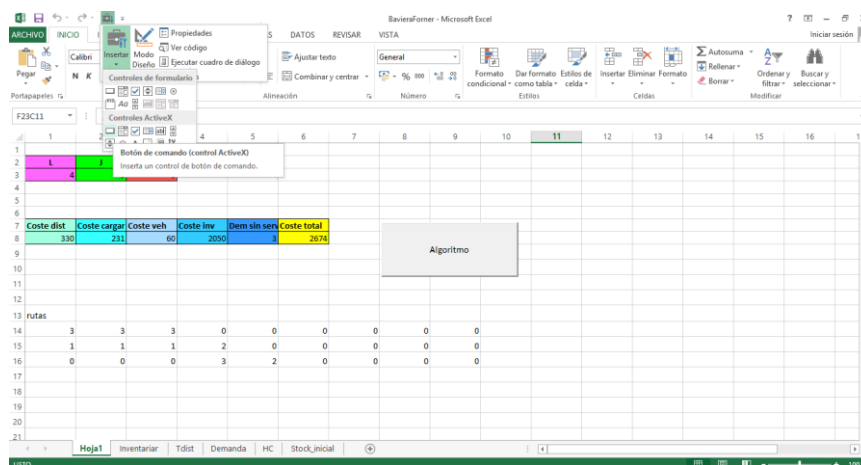


Imagen 18. Cómo crear una macro desde cualquier versión anterior a Microsoft Office Excel 2010

Cuando se crea una macro, Visual Basic for Applications (VBA) aparece con las tres ventanas siguientes:

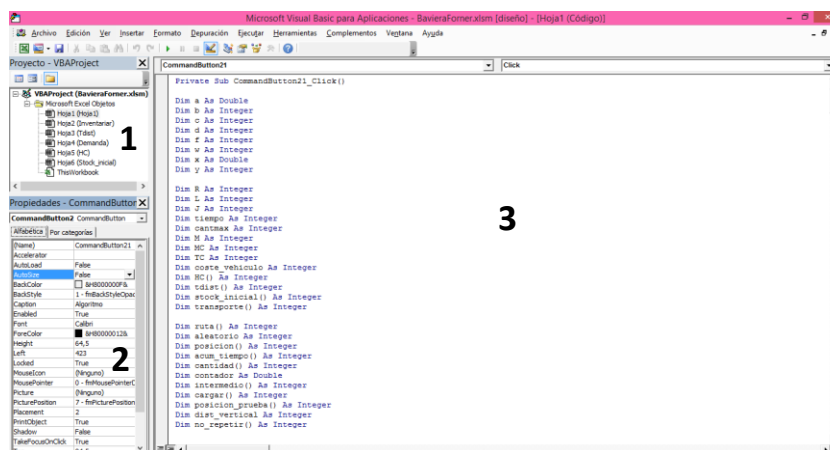


Imagen 19. Ventanas de la macro.

La primera ventana muestra el Proyecto, es decir, todas las hojas que forman este libro ya que se puede programar en todas ellas, aunque en este caso, solo se haya programado en una de ellas.

La segunda ventana muestra las propiedades del botón de la macro. Aquí se pueden cambiar las propiedades del botón, como son: nombre, el tipo de letra y su color.

Por último, mediante la tercera pantalla se puede empezar a programar.

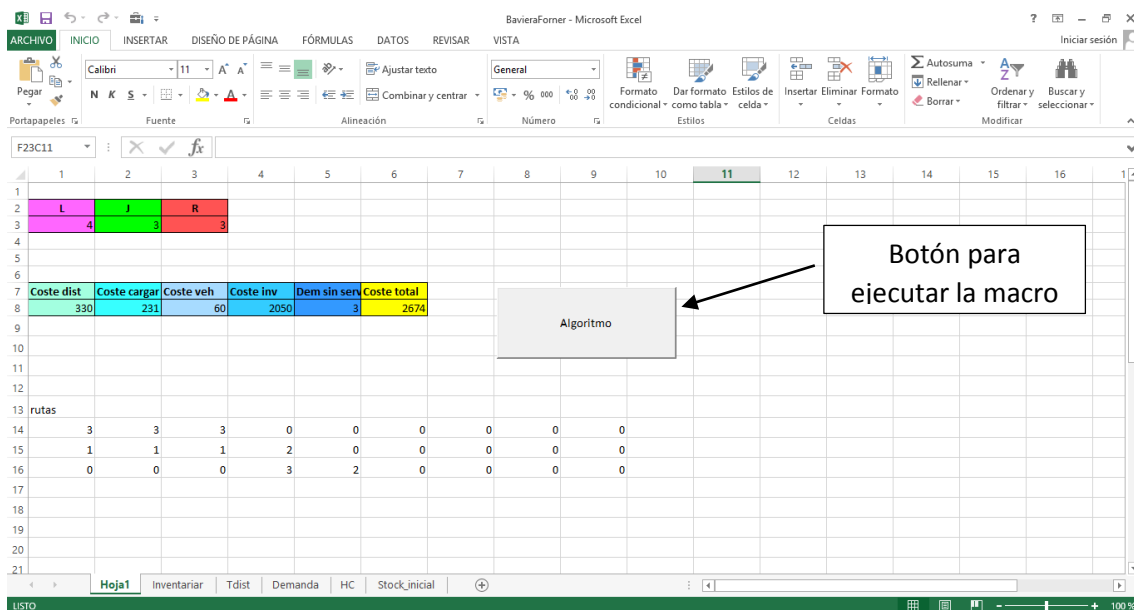


Imagen 20. Pantalla principal de Excel

5.1.3. Características de la herramienta

Una vez se sepa cómo funciona la herramienta y antes de empezar a programar, hay que declarar las variables que se van a utilizar. Para ello, hay que tener en cuenta qué tipo de datos va a tener cada variable. Existen diversos tipos de variables y los más utilizados son los siguientes:

Tipo	Asignación de almacenamiento	Intervalo de valores
Char	2 bytes	0 a 65535
Short	2 bytes	-32.768 a 32.767
Integer	4 bytes	-2.147.483.648 a 2.147.483.647
Single	4 bytes	-3,4028235E+38 a -1,401298E-45 para valores negativos; 1,401298E-45 a 3,4028235E+38 para los valores positivos
Long	8 bytes	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 (9,2...E+18 †)
Double	8 bytes	-1,797693E+308 a -4,940656E-324 † para valores negativos; 4,940656E-324 a 1,7976930E+308 † para valores positivos
Decimal	16 bytes	0 a +/-7,9228162514264337593543950335 con 28 posiciones a la derecha del decimal
String	En función de la plataforma de implementación	0 a 2.000 millones de caracteres Unicode aprox.

Tabla 10. Tipos de variables y sus rangos.

Se elegirá el tipo de variable dependiendo del valor que vaya a tener ésta. Siempre se intentará utilizar el tipo que menos memoria ocupe dado que cuanto menos memoria ocupe, más rápido se podrá ejecutar el programa. Cuando los problemas no son de demasiada envergadura y ya se sabe de antemano que es poco probable que las variables vayan a tomar valores excesivamente grandes, siempre se debería empezar declarando las variables numéricas y de valores enteros con el tipo "Integer". Esto es beneficioso porque aunque después alguna variable conlleve valores de rangos más amplios, el programa avisará al usuario de ello y se podrá cambiar la declaración. Esto significará que todas las variables están ajustadas a sus mínimos valores, lo cual implica una menor ocupación de la memoria.

Cuando las variables sean numéricas pero con decimales, se utilizará el tipo "Decimal" para declararlas y para las variables que utilicen caracteres en lugar de números, se utilizará el tipo "Char".

Además, hay que tener en cuenta para cualquier vector o matriz declarado, hay que definir su tamaño. Por ello, se utiliza "ReDim". Su función es redimensionar la variable a los tamaños que el usuario exigirá en ese momento.

Una vez todas las variables se hayan declarado y los vectores y matrices dimensionados, se puede proceder a programar. Para poder hacerlo, hay que tener en cuenta los siguientes pasos:

1. Análisis de requisitos: determina qué es lo que se está buscando con la creación del programa. Esta fase debe estar bien definida puesto que más tarde no va a poder modificarse.
2. Diseño del programa: creación de los algoritmos que se van a utilizar durante el programa.
3. Codificación: transformación del código en un lenguaje determinado de programación.
4. Pruebas: realización de pruebas para comprobar que el programa funciona como se espera, cumpliendo con los requisitos establecidos al principio.
5. Verificación: el cliente prueba el programa comprobando que funciona cumpliendo con sus expectativas.
6. Mantenimiento: se encarga de solucionar los posibles errores o deficiencias del programa creado. Existe la posibilidad de que algunos de ellos necesiten reiniciar el ciclo de vida ya que no están bien definidos.

En el siguiente diagrama se pueden observar los pasos a seguir a la hora de programar:



Diagrama 5. Ciclo en cascada. (Winston W. Royce en 1970 y posteriormente revisada por Barry Boehm)

5.2. Algoritmos utilizados

Algoritmo:

Un algoritmo es una secuencia de instrucciones que, partiendo de un estado básico o inicial y por medio de una sucesión de pasos, consigue llegar a obtener la solución a un problema.

Los algoritmos son independientes de los lenguajes de programación. No importa el lenguaje que se esté utilizando (C++, Visual Basic for Applications...), el algoritmo se puede construir y después utilizarse con los diferentes lenguajes de programación modificando únicamente el lenguaje en el que está escrito.

Partes de las que está constituido un algoritmo:

1. Datos de entrada: estos son los datos que proporcionará el usuario y son los datos que se necesitan para empezar a ejecutar el algoritmo.
2. Proceso: secuencia de instrucciones que va ejecutando el algoritmo.
3. Datos de salida: valores que devuelve el algoritmo una vez ha terminado de ejecutarse.

Características de los algoritmos:

- Debe terminar con un resultado: debe proporcionar una solución al problema.
- Preciso: cada paso debe ser definido con exactitud y sin dar lugar a ambigüedades.
- Definido: el algoritmo dará el mismo resultado siempre con los mismos datos de entrada, sin importar cuantas veces se ejecute de nuevo.
- Finito: un algoritmo debe de tener un número limitado de pasos.

Algoritmo de búsqueda local:

Un algoritmo de búsqueda local trata de encontrar mejores soluciones que la que ya tiene, buscando en el vecindario. Este algoritmo parte una solución inicial (que puede haberse obtenido aleatoriamente o mediante otro algoritmo) y a partir de ahí, mediante transformaciones, busca nuevas soluciones. Si la nueva solución encontrada es mejor que la que ya tenía guardada, ésta la sustituirá y en caso contrario, seguirá haciendo más permutaciones. El algoritmo dejará de ejecutarse cuando haya terminado con todas las transformaciones.

Añadiendo a lo anterior, si el programa hace tantas iteraciones como permutaciones existan, entonces el algoritmo encontrará la solución óptima. En caso contrario, encontrará como mínimo una buena solución, es decir, de entre todas las soluciones escogerá la mejor de ellas. Por ejemplo, si se quisiera encontrar el camino más corto para poder satisfacer la demanda de tres localizaciones, las diferentes combinaciones posibles serían: 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2 y 3-2-1. Si el algoritmo probase los 6 distintos escenarios, entonces encontraría la mejor solución. Si en cambio sólo estudiase 4 de los 6 posibles casos, entonces puede o no encontrar la solución óptima.

Los algoritmos de búsqueda local cuentan principalmente con dos ventajas:

1. Usan muy poca memoria
2. Son capaces de encontrar buena soluciones en grande espacios para los cuales los algoritmos sistemáticos son inadecuados.

En este caso, para buscar soluciones en el vecindario se ha decidido hacerlo buscando soluciones aleatoriamente. El algoritmo lo que hace es que va probando diferentes combinaciones de la solución posible (en este caso el orden en el que se servirá a cada localización) y lo hace de manera aleatoria. Cuando encuentre una solución mejor que la que ya tenía, se la guardará, en caso contrario, la desechará y seguirá buscando nuevas soluciones.

5.3. Análisis de requisitos

A la hora de crear el algoritmo en VBA, éste va a definirse de manera más completa a la que se ha definido en el modelo matemático, principalmente porque éste va a poder realizar más acciones, encontrando otras soluciones que el modelo matemático no proporciona.

En primer lugar, aquí se tendrá en cuenta un coste más. Este coste será llamado “demanda sin servir”. Este coste se ha considerado puesto que sería muy negativo para la empresa el no disponer de un material a tiempo. El no disponer de un material a tiempo acabaría perjudicando a toda la línea (aunque sólo fuese un puesto de trabajo el que se quedara sin satisfacer) dado que esto podría resultar en una parada de la línea. Evidentemente, esta situación se quiere evitar a toda costa y por ello, se le pone un coste elevado por cada unidad que no se sirva a tiempo. Además, un retraso aquí significa un retraso para el cliente, con lo cual, el nivel de satisfacción del mismo disminuiría.

Añadiendo a lo anterior, lo que se consigue imponiendo un coste elevado a cada unidad no servida a tiempo es que el algoritmo evitará que el número de éstas aumente, intentando siempre llegar a tiempo para servir todos los puestos, o buscando rutas alternativas para poder llegar a tiempo aunque esto aumente el coste de inventario.

Mediante el algoritmo, además de calcular los costes totales para un cierto número de carretillas, también calculará las rutas que seguirán estas carretillas. Lo que el algoritmo tratará de hacer será: conociendo la demanda que debe servir y el número de carretillas que dispone, ver por qué sitios deberá pasar con cada carrito y en qué momento para intentar siempre llegar en el momento preciso, es decir, ni demasiado pronto pues esto acabará en elevados costes de inventariar e incluso que no haya suficiente espacio para depositar las piezas, pero tampoco tarde para que el operario no tenga que parar la línea.

Hay que tener en cuenta que durante la realización del algoritmo, también se realizarán varias pruebas para determinar cuál es la cantidad de carretillas ideal a tener, al mínimo coste.

5.4. Diseño del Programa y Codificación

Diseño interior

A continuación se van a citar los diferentes apartados que forman parte del programa realizado en Visual Basic for Applications. Además, se explicará en qué consiste cada uno de ellos para poder entender con mayor facilidad qué es lo que hace la herramienta en cada momento. En este apartado también se definirá qué representa cada variable, pues se han utilizado muchas de ellas y sin saber qué están representando es imposible deducir qué está haciendo el programa. Este último paso es muy importante puesto que a la hora de programar cada programador lo hace a su manera y por lo tanto, llama a las variables según crea conveniente. Una explicación sobre las mismas hará que se entienda el programa con una mayor facilidad.

Declaración de variables:

Como se ha explicado anteriormente, el primer paso para programar en VBA es declarar las variables según el tipo que sean. En este caso, aunque se ha visto que hay muchos tipos de variables, sólo se van a utilizar 2. La primera es "Integer". La mayoría de las variables van a ser del tipo "Integer" debido a que los valores que van a asumir van a ser numéricos y no de grandes tamaños. El siguiente tipo de datos van a ser "Single". Estas variables declaradas como "Single" son todos los costes, además de otras variables que vayan a almacenar valores mayores a los permitidos por "Integer".

La forma en la que se evalúa el programa es mediante unas variables denominadas *costes*. Estos costes, a medida que se van teniendo más localizaciones, más componentes y más vehículos, adquieren valores mayores, con lo cual, se llega a un punto en que no es suficiente declararlos como "Integers" y por ello deben ser declaradas como "Single". Las variables "Single" son los siguientes tipos de datos que menos espacio ocupan, después de "Integers", pero que pueden tomar valores más elevados. En el caso en el que se supiera que el número de localizaciones y componentes no vaya a ser muy elevado, declarándolos como "Integer" sería suficiente.

Explicación de las variables:

Índices: a, b, c, d, f, w, x, y → Estas variables se utilizan para avanzar en los vectores y las matrices

Valor de las variables:

Variable	Significado
R	Número máximo de vehículos
L	Número de localizaciones máximas a las que servir
J	Máximo número de componentes diferentes
tiempo	Periodo de tiempo en el que se desarrolla el proyecto
cantmax	Número de unidades máximas que se pueden transportar en un vehículo
M	Número muy grande
MC	Coste por unidad transportada
TC	Coste por unidad de distancia
coste_vehículo	Coste por vehículo
HC()	Coste unitario de almacenar cada componente "j"
tdist	Tiempo (medido en distancias) de desplazamiento entre localizaciones
stock_inicial	Stock inicial de cada componente "j" en cada localización "l"
transporte	Demanda de cada componente "j" en cada localización "l"

Tabla 11. Nomenclatura. Valor de las variables.

Una vez se han declarado todas las variables, se procede a definir el valor de las variables. En este caso, el valor de las variables serán los datos de entrada que vendrán dados por el cliente. Las variables: localización (L), componente (J) y carretillas (R) variarán de programa en programa, dado que es lo que se está intentado evaluar, es decir, el saber cómo afecta el número de carretillas dependiendo de las diferentes cantidades de localizaciones y componentes que hayan. En cambio, el resto de variables, como son por ejemplo la cantidad máxima (cantmax) o el tiempo máximo que puede tardar en hacer el recorrido una carretilla (tiempo), serán siempre fijas, puesto que son impuestas por el cliente.

Redimensionar los vectores y matrices:

Después de esto, se empieza a definir el tamaño de los vectores y las matrices. Al principio del programa ya se han declarado estas variables, pero lo que no se ha hecho es precisar las dimensiones que van a tener. Por ello, aquí se dimensionarán. Añadiendo a lo anterior, no todas las variables se van a poder dimensionar puesto que necesitan hacerlo durante el programa. Esto se realiza para no dar un tamaño más grande del necesario a las variables ya que si se dimensionaran ahora, variables como intermedio podría llegar a tener hasta un tamaño 500 veces mayor de lo que debería.

Leer los datos de entrada:

Una vez ya se tengan las dimensiones de los vectores y las matrices, se define el valor que van a tener éstas. Para simplificar este proyecto, los datos de entrada de los vectores y matrices, se encuentran en otras hojas del mismo libro de Excel. Estos datos serán:

- *Tdist*: distancia que hay entre una localización y otra
- *Demanda*: cantidades a satisfacer por el cliente
- *HC*: coste asociado a transportar cada componente debido a que tienen tamaños diferentes
- *Stock_inicial*: stock con el que se encuentra en este momento cada localización dentro de la empresa

Inicializar las variables:

Por último, antes de empezar a programar, se inicializarán todas las variables, vectores y matrices a cero. Esto se hace para que después no haya errores dado que hay veces que el programar podría darle un valor a una variable que en realidad no es el correcto ya que lo tiene grabado en la memoria de otra ejecución. Algunas de las variables no se inicializarán a cero en este momento, si no durante el transcurso del programa dado que el mismo se va a ejecutar varias veces para así poder obtener una mejor solución, si no se inicializaran cada vez que se repite el programa, el valor de las variables aumentaría en cada ejecución y no obtendrían el valor que deberían.

Ejecutar el programa 100 veces:

Variable	Definición
sol_coste_total	mejor coste_total encontrado

Tabla 12. Nomenclatura. Ejecutar el programa 100

El programa se va a ejecutar 100 veces. Esto se va a hacer para intentar conseguir el mejor resultado posible. En este programa se va a trabajar con heurísticas donde se desea obtener la mejor solución posible, aunque esto no quiere decir que se vaya a encontrar la solución óptima. Es por ello por lo que al ejecutar el programa varias veces, significa que se obtienen diferentes soluciones, donde algunas serán mejores que otras y se puede alcanzar una buena solución. En este caso, para evaluar qué solución es mejor que otra, se ha creado la variable *sol_coste_total*. Antes de ejecutar por primera vez el programa, la variable es inicializada a un valor muy elevado para que cuando el programa se ejecute la primera vez, el primer valor al que se comparará *sol_coste_total* será el del primer coste. Si esto no se hiciera y el valor de *sol_coste_total* fuera cero como el resto de variables que se inicializan al principio, el programa se ejecutaría todas las veces pero no guardaría ninguna respuesta puesto que cualquier solución ya tendría algún coste, y éste siempre sería mayor a cero.

El programa se ejecutará 100 veces puesto que se ha considerado que son suficientes veces para conseguir una buena respuesta y si se repitiera muchas más veces, tardaría mucho más en encontrar la solución, puesto que cuantas más veces se repita, más complejo será el programa y mayor será el tiempo de respuesta y además, seguramente, la solución que encuentre ya la habría encontrado con anterioridad puesto que las soluciones posibles que existen no son demasiadas para el tamaño que tienen los datos de entrada. Si en cambio se tratara de un programa donde existen millones de soluciones, cuantas más veces se ejecute el programa, más probabilidades habrán de encontrar mejores soluciones.

Crear una ruta provisional aleatoria:

Variable	Significado
ruta	Vector que representa el trayecto que va a seguir cada carretilla "r" para llegar a todas las localizaciones "l"
aleatorio	obtiene un valor aleatorio entre cierto rango definido (entre 1 y (L-1))
pos_aleat	almacena el número de localización que haya salido en la variable aleatorio

Tabla 13. Nomenclatura. Crear una ruta provisional aleatoria.

En esta primera parte, se va a crear una ruta provisional aleatoria. Esta ruta será la base desde la que partirá el problema. Hay que tener esta ruta provisional puesto que el programa debe empezar a actuar sobre una solución y aquí es dónde se le proporciona una. Se podría empezar siempre con una misma ruta, pero para que el programa sea más eficiente y encuentre más opciones, se empezará con una ruta aleatoria en cada ciclo.

Las opciones de esta ruta son las diferentes localizaciones, es decir, que si hubiesen por ejemplo 4 localizaciones (L) las opciones serían: 1, 2 y 3. Siempre se tendrá una localización menos (L-1) dado que hay que tener en cuenta que siempre se partirá del "warehouse" y esa será la localización 0. De entre estas opciones, aleatoriamente se elegirá el orden que deberán seguir. Esto se hace mediante la función Random "Rnd".

Cantidad a transportar en cada viaje:

Variable	Significado
acum_tiempo	vector que acumula el tiempo que se tarda en realizar una ruta por carretilla "r"
cantidad	vector que acumula la cantidad que va cargando cada carretilla "r"
contador	número de carretillas necesarios para que cada uno haga únicamente una vuelta
intermedio	matriz que muestra las localizaciones "l" a las que va cada carretilla "r"
cargar	matriz que muestra la cantidad a transportar de cada carretilla "r"
posición	matriz que muestra la posición de la localización "l" y componente "j" desde la que se va a transportar
columnas	Se utiliza para calcular num_columnas
num_columnas	Número de rutas máximo al que acudirá una carretilla

Tabla 14. Nomenclatura. Cantidad a transportar en cada viaje

Sabiendo la demanda que va a haber y que hay un límite tanto de tiempo (cada carretilla no podrá tardar más que 20 unidades de tiempo) como de cantidad (cada carretilla sólo puede transportar 10 unidades por viaje) empezará a cargar por cada ruta, la mayor cantidad posible de material. Este proceso se realiza siguiendo el orden de rutas proporcionado en el apartado anterior. Una vez la carretilla esté llena o vaya a tardar más que el tiempo máximo, otra carretilla continuará con la ruta. Es por ello, por lo que se crean las siguientes matrices:

- Intermedio: guarda las localizaciones que seguirá cada carretilla. Cada nueva fila representa una nueva carretilla.
- Cargar: cantidades que transporta por cada localización y cada carretilla
- Posición nueva: esta matriz servirá más adelante para saber exactamente desde qué localización y qué componente se está transportando

La variable contador es la equivalente al número de carretillas. En este momento, la variable contador representa la cantidad de carretillas que se necesitarían para satisfacer toda la demanda sin que ninguna carretilla hiciera más de un recorrido.

Multiplicador:

Variable	Significado
multiplicador	Número de vueltas que deberá realizar al menos una carretilla "r" para poder transportar todo el material demandado
auxiliar1	Variables que se utilizan para la obtención de la variable multiplicador
redondeo	

Tabla 15. Nomenclatura. Multiplicador.

Esta variable es creada para saber el número de recorridos que debería hacer cada carretilla sabiendo que hay un número fijo de carretillas y por tanto, no se pueden utilizar todos los que se quieran.

Que sólo haya como máximo R número de rutas:

Variable	Significado
vector_intermedio	vector que guarda los valores de la matriz_intermedio
vector_cargar	vector que guarda los valores de la matriz_cargar
vector_posición	vector que guarda los valores de la matriz_posición
nuevo_intermedio	matriz que representa las rutas que va a seguir cada carretilla "r", sabiendo que carretillas $\leq R$
nuevo_cargar	matriz que representa los valores de las cantidades que se van a transportar, sabiendo que número de carretillas $\leq R$
nuevo_posición	matriz que representa las posiciones de cada localización "l" y componente "j", sabiendo que carretillas $\leq R$

Tabla 16. Nomenclatura. Que sólo haya como máximo R número de rutas (I).

Este apartado se divide en dos partes. En la primera parte se reduce el número de contador al número de carretillas que haya disponibles. Evidentemente, si el número de carretillas es igual o menor a "R", no hará falta que el programa haga nada. En caso contrario, las 3 matrices principales son transformadas en unas nuevas matrices pero con el número de carretillas=R.

Variable	Significado
mejor_matriz	Matriz que representa las rutas que va a seguir cada carretilla “r” con el menor tiempo posible
mejor_cargar	Matriz que cantidades que se van a transportar con el menor tiempo posible
mejor_posición	Matriz que representa las posiciones de cada localización “l” y componente “j” con el menor tiempo posible
guardar_aleat	obtiene un valor aleatorio entre cierto rango definido (entre 1 y (L-1))
nuevo	almacena el número de localización que ya haya salido mediante la variable guardar_aleat
mín_tiempo	tiempo más corto en el que se realiza la ruta
suma_tiempo	tiempo total que tarda una carretilla “r” en realizar una ruta

Tabla 17. Nomenclatura. Que sólo haya como máximo R número de rutas (II).

En la segunda parte trata de encontrar la mejor solución posible por cada carretilla. Sabiendo las diferentes localizaciones a las que debe ir cada uno en cada viaje y basándose en el tiempo que se tarda de ir a una localización a otra, se probarán diferentes combinaciones de rutas dentro de cada carretilla y se quedará con la que tarde menos tiempo. A partir de aquí, las 3 matrices principales pasarán a llamarse igual que antes, pero con el prefijo “mejor” dado que son la mejor solución encontrada.

Inventariar:

Variable	Significado
tiempo_mío	vector que acumula el tiempo de cada carretilla durante toda su ruta
llegada	Momento en el tiempo en el que llega cada unidad de material
llegada_acum	Sumatorio tanto de las unidades de materiales que llegan como de las que ya se encuentran la localización “l” en cada momento del tiempo

Tabla 18. Nomenclatura. Inventariar.

Antes de poder inventariar, la matriz “tiempo_mío” es creada para que almacene el momento en el tiempo en el que se encuentra la carretilla al ir de una localización a otra. Una vez se sabe cuándo estaría cada carretilla en cada localización, se crea la matriz variable. Esta variable dependerá del tiempo y no de las carretillas como se ha utilizado durante el desarrollo de esta herramienta. Ahora esta matriz es independiente del número de carretillas y de qué carretilla hace qué ruta puesto que lo que realmente le interesa es saber cuándo y qué componentes llegan en cada unidad de tiempo. Además, hay que tener en cuenta que lo que no se ha consumido en una unidad de tiempo, continua estando en la siguiente unidad de tiempo, y por ello existe la matriz “llegada_acum”.

Por último, la variable “demanda_sin_servir” es introducida. Siempre que lo que se está transportando mediante las carretillas llegue al mismo tiempo o antes de lo que es demandado, esta variable permanecerá en cero. En cambio, cuando lo que se transporte sea menor a lo demandado, la variable “demanda_sin_servir” irá sumando toda la demanda que no ha podido ser satisfecha.

Calcular costes:

Variable	Significado
coste_dist	Sumatorio del coste que conlleva ir de una localización “l” a otra
total_cargar	Coste de transportar los componentes “j” entre las localizaciones “l” por todas las rutas
coste_inv	Coste por tener piezas almacenadas sin consumir
demanda_sin_servir	Demanda que no ha podido ser satisfecha
coste_total	Suma de todos los costes

Tabla 19. Nomenclatura. Calcular costes.

En este apartado se calcularán tres de los cinco costes (puesto que los otros dos restantes ya se han calculado durante el transcurso del programa). El significado de estos costes ya se ha explicado con el modelo matemático anterior.

Que se quede con la mejor solución posible:

Variable	Significado
sol_ruta	Con el menor tiempo y coste posible, mejor ruta encontrada
sol_coste_dist	Con el menor tiempo y coste posible, mejor coste_dist encontrado
sol_total_cargar	Con el menor tiempo y coste posible, mejor total_cargar encontrado
sol_coste_inv	Con el menor tiempo y coste posible, mejor coste_inv encontrado
sol_contador	Con el menor tiempo y coste posible, mejor contador encontrado
sol_demanda_sin_servir	Con el menor tiempo y coste posible, mejor demanda_sin_servir encontrada
Sol_num_columnas	Con el menor tiempo y coste posible, mejor num_columnas encontrado
Sol_multiplicador	Con el menor tiempo y coste posible, mejor multiplicador encontrado
Sol_mejor_matriz	Con el menor tiempo y coste posible, mejor mejor_matriz encontrado

Tabla 20. Nomenclatura. Que se quede con la mejor solución posible.

Como se ha mencionado anteriormente, aquí es donde se decide si la solución obtenida en esta ronda es mejor que cualquier otra que ya haya salido. Para poder identificar qué solución es mejor que cualquier otra, la variable “sol_coste_total” almacenará el mejor valor encontrado, es decir, a media que se ejecute el programa, siempre que “sol_coste_total” > “coste_total”, se guardará la nueva solución, siendo “sol_coste_total” = “coste_total”. Si por el contrario “coste_total” tuviera un coste mayor que “sol_coste_total”, este valor se ignoraría y se seguiría ejecutando el programa sin importar esta vez el valor obtenido de la variable.

Diseño exterior

Una vez finalizado el diseño interior de cálculos y la codificación se procede a describir el diseño exterior que será la parte interactiva del mismo que utilizará el usuario de este programa.

La herramienta desarrollada en Visual Basic se presenta con el siguiente formato:

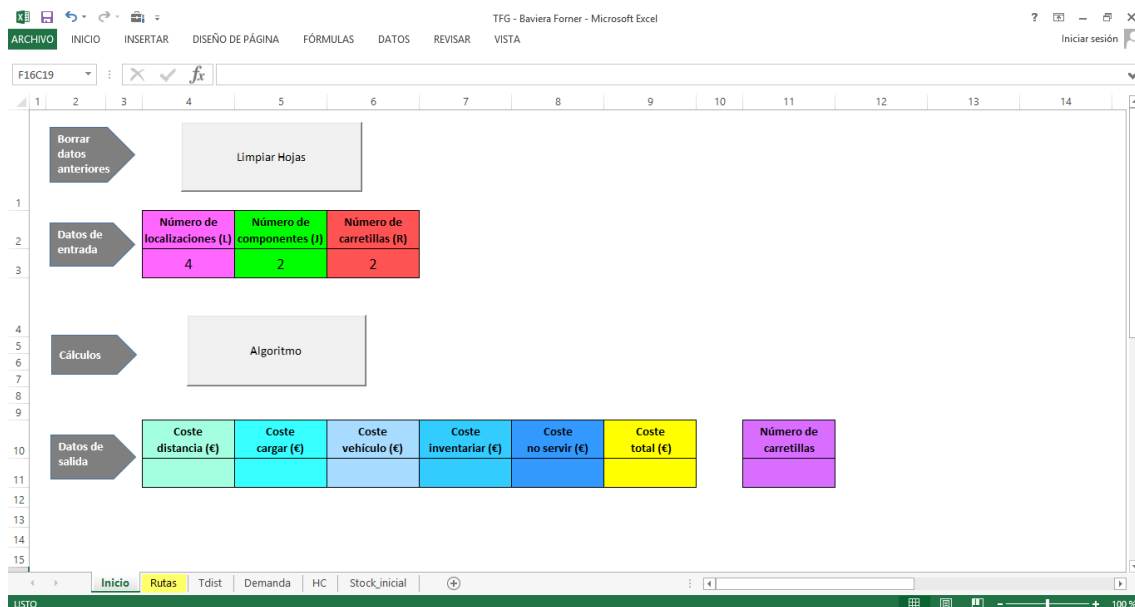


Imagen 21. Diseño exterior.

En la primera hoja del archivo “Excel” se encuentra la pestaña “Inicio”. En ésta podemos observar secuencialmente en sentido descendente:

- Borrar datos anteriores: esta macro “Limpiar hojas” borra el contenido de todas las celdas de salida. Esta macro debe ejecutarse antes de ejecutar “Algoritmo”.
- Datos de entrada: datos sobre Número de localizaciones, Número de componentes y Número de carretillas, proporcionados por el usuario y que se irán variando durante esta fase de experimentación.
- Cálculos: ejecución de la macro “Algoritmo” que resuelve el problema.
- Datos de salida: resultados que nos proporciona el programa en relación a los diferentes costes considerados, el coste conjunto y el número de carretillas.

En la siguiente pestaña de la hoja, llamada “Rutas” se encuentran las distintas rutas que deberá recorrer cada carretilla para poder satisfacer las necesidades de todos los puestos de trabajo.

Por último, en el resto de hojas “Tdist, Demanda, HC, Stock_inicial” se muestran los datos de entrada que permanecerán fijos durante toda la experimentación.

El programa está diseñado para un máximo de 50 localizaciones (L) y 20 componentes (J) distintos.

5.5. Pruebas y Verificación

Experimentación I: problemas de pequeñas dimensiones

Experimento 1: L=4, J=2, R=2

La primera situación que se planteará será la misma que a la hora de resolver el modelo matemático en el punto 4.3. del trabajo, es decir, L=4, J=2, R=2.

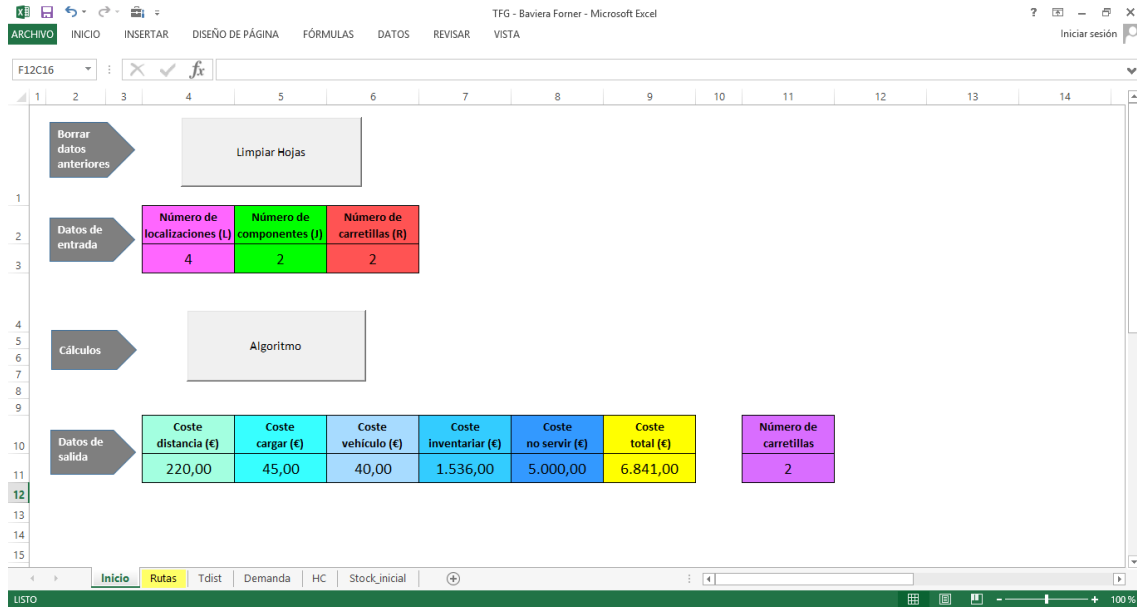


Imagen 22. Experimento 1. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	1	0	1	2	0	1	2	0
Carretilla 2	2	0	3	0	2	3	0	

Imagen 23. Experimento 1. Rutas.

Representación de la solución:

Carretillero 1. Ruta 1:

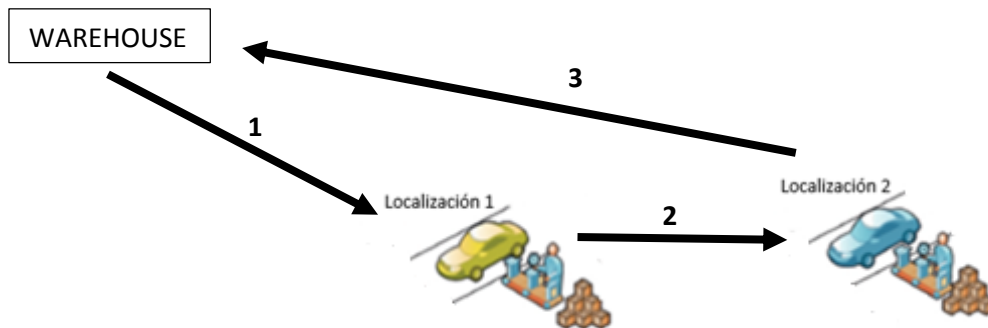


Imagen 24. Experimento 1. Carretillero 1. Ruta 1.

Carretillero 1. Ruta 2:

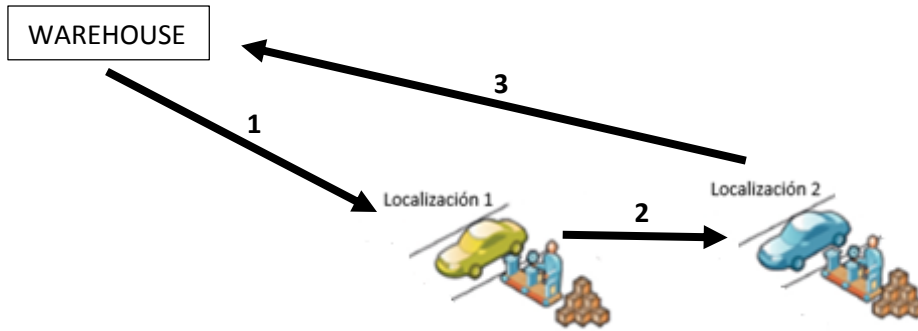


Imagen 25. Experimento 1. Carretillero 1. Ruta 2.

Carretillero 2. Ruta 1:

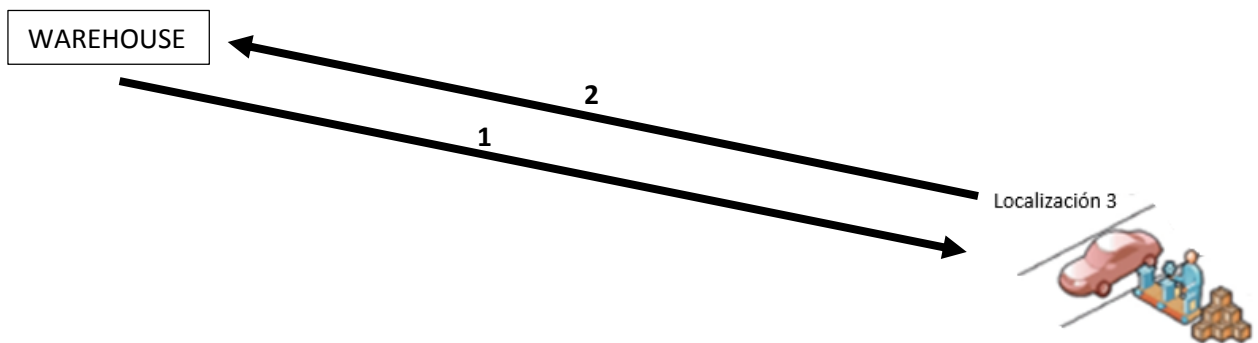


Imagen 26. Experimento 1. Carretillero 2. Ruta 1.

Carretillero 2. Ruta 2:

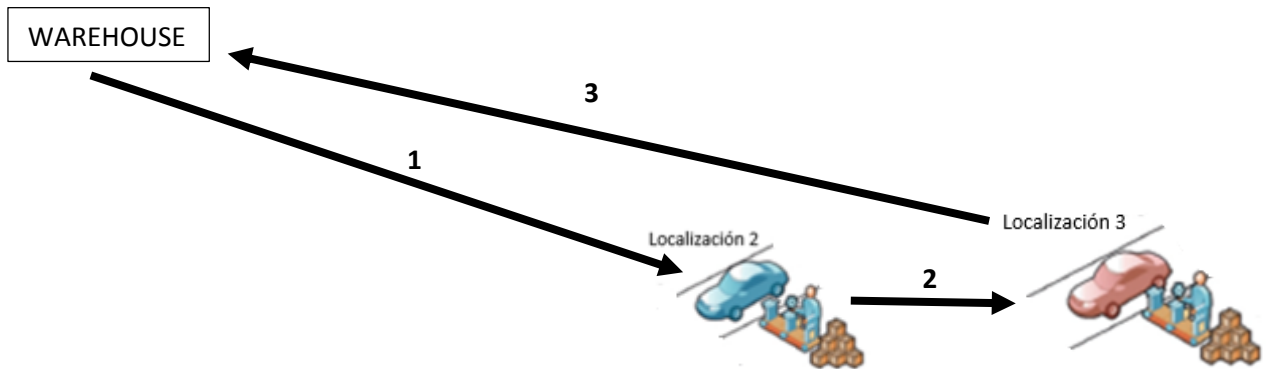


Imagen 27. Experimento 1. Carretillero 2. Ruta 2.

Mediante este primer experimento se puede observar que con la solución obtenida no se sirve toda la demanda a tiempo. Ante esta situación, caben dos alternativas, se aumenta el número de carretillas o se atiende tarde la demanda. Evidentemente, la primera solución es preferible ya que el coste de tener una carretilla más no es comparable al hecho de no servir a tiempo la demanda y tener que parar la línea. Por ello, en la siguiente ejecución se aumentará el número de carretillas a tres.

Experimento 2: L=4, R=2, J=3

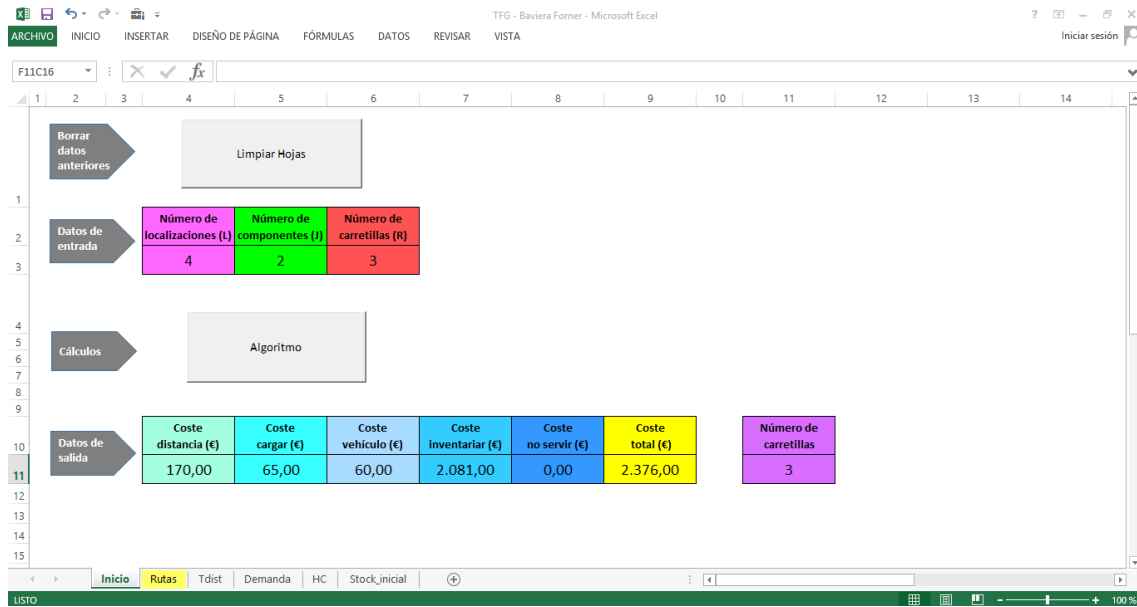


Imagen 28. Experimento 2. Inicio

Localizaciones (L=0=warehouse)

Carretilla 1	1	0	1	2	0	2	3	0
Carretilla 2	0	3	0					
Carretilla 3	0	1	2	0				

Imagen 29. Experimento 2. Rutas

En esta ocasión, mediante las rutas sugeridas por la herramienta, sí se satisface a tiempo la demanda de todas las localizaciones. Si se llegara a tener esta situación en la vida real, habría que comprar una carretilla más; pues aunque el coste de adquisición de una carretilla es elevado, es evidente que aun así resulta beneficioso en comparación a las pérdidas que ocasiona una parada de línea por falta de suministro y no servir a tiempo a los clientes.

Experimento 3: L=4, J=2, R=4

Mediante este experimento se tratará de ver si resulta más beneficioso el seguir aumentando el número de carretillas, o si con tres sería suficiente (dado que con éstas ya se satisface toda la demanda).

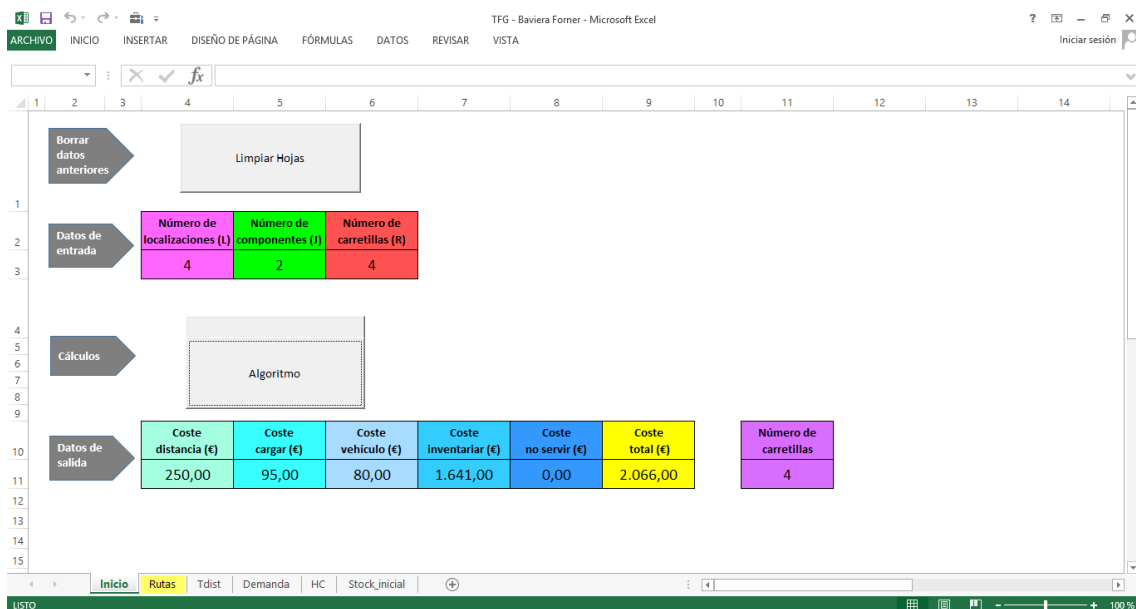


Imagen 30. Experimento 3. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	1	0	1	0	2	0
Carretilla 2	2	0	3	2	0	0
Carretilla 3	3	0	2	1	0	0
Carretilla 4	4	0	3	2	0	0

Imagen 31. Experimento 3. Rutas.

En este caso, el coste es de 2.066,00€, menor que en la situación anterior (2.376,00€). Este coste menor se debe a que ahora el material llegará en un tiempo más ajustado, con lo que se inventariarán menos unidades. Esta es la situación idónea, donde se inventaria lo menor posible pero a la vez se sirve la demanda a todos los puestos de trabajo. El único problema es que en este caso, si no se dispusiera de otra carretilla en la fábrica, habría que comprar una carretilla más y éstas, aunque sean de segunda mano, tienen un coste mínimo de 6.500€ que es menor al ahorro que supondría esta situación. Por lo tanto, se utilizarían tres carretillas para satisfacer la demanda de 4 localizaciones y 2 componentes diferentes.

Experimento 4: L=4, J=4, R=3

Durante esta experimentación se incrementará el número de componentes en cada localización, pues se podría dar la ocasión de que debido a una modificación en el automóvil hubiera que añadir nuevas piezas a un modelo y por lo tanto habría que suministrar una mayor cantidad que de las que se estaban suministrando.

Como mediante el experimento anterior se ha visto que se han necesitado tres carretillas para satisfacer la demanda, se empezará con esa situación.

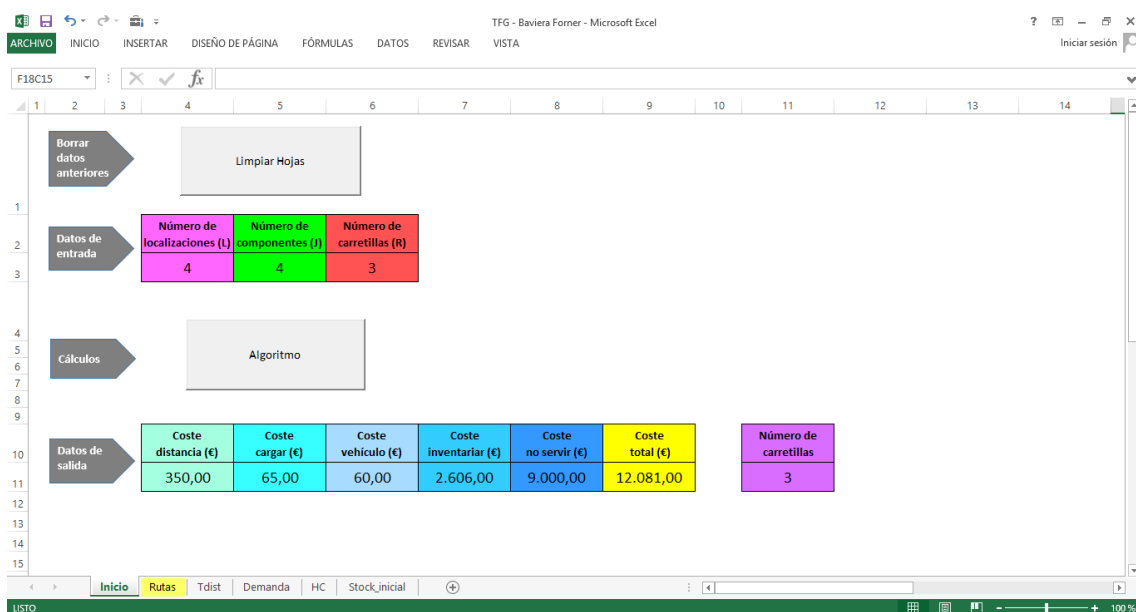


Imagen 32. Experimento 4. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	3	0	2	3	0
Carretilla 2	0	1	2	0	3	0
Carretilla 3	0	2	3	0	1	2

Imagen 33. Experimento 4. Rutas.

Como se puede observar, en esta situación no se consigue satisfacer la demanda. Como se ha mencionado anteriormente, esta situación es la que se quiere evitar, por lo tanto, se incrementará el número de carretillas.

Experimento 5: L=4, J=4, R=4

En este experimento se aumentará el número de carretillas para ver si así se consigue satisfacer toda la demanda.

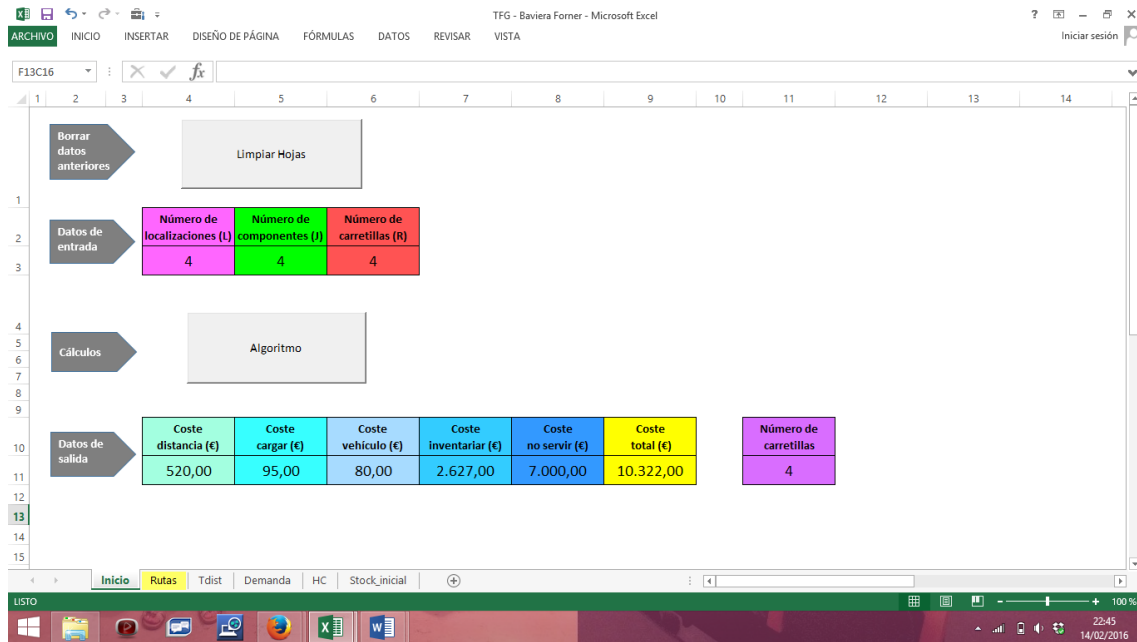


Imagen 34. Experimento 5. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	1	0	1	0	2	0
Carretilla 2	0	1	3	0	1	0	0
Carretilla 3	0	2	0	3	0	0	0
Carretilla 4	0	2	0	3	2	0	0

Imagen 35. Experimento 5. Rutas.

De nuevo, mediante esta situación tampoco se llega a servir a tiempo. Por ello, se incrementará el número de carretillas hasta cinco para ver si de esta forma se consigue satisfacer toda la demanda.

Experimento 6: L=4, J=4, R=7

El programa se ejecutará tantas veces como sea necesario hasta obtener un coste por demanda sin servir de cero. Esto es equivalente a que se sirva toda la demanda de los diferentes puestos de trabajo a tiempo.

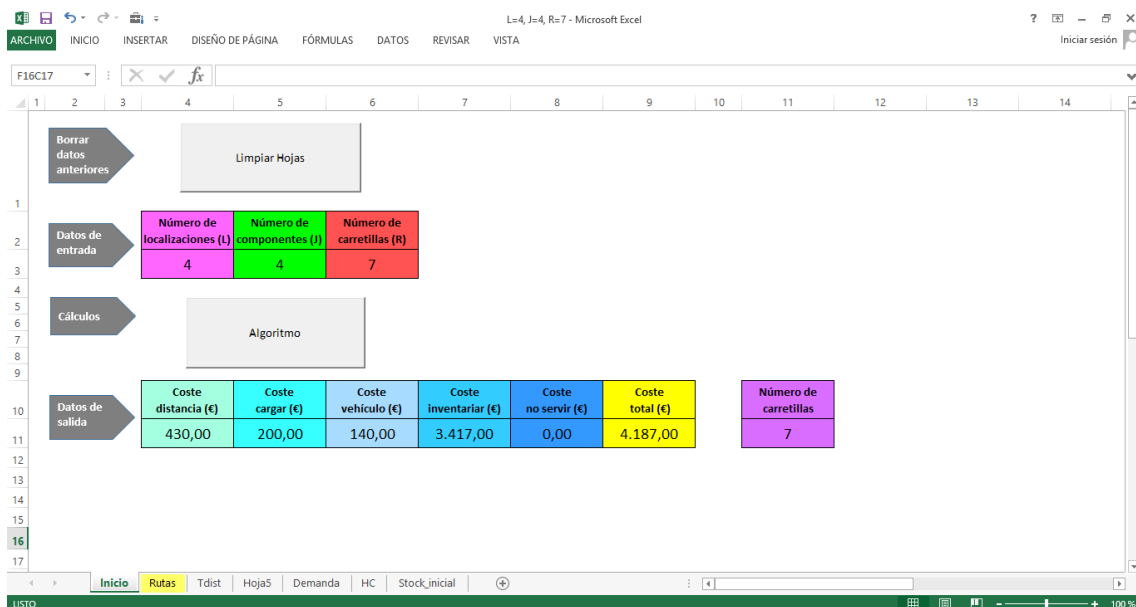


Imagen 36. Experimento 6. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	1	0	3	0
Carretilla 2	0	1	2	0	3
Carretilla 3	0	2	3	0	
Carretilla 4	0	3	1	0	
Carretilla 5	0	1	0	0	
Carretilla 6	0	1	2	0	
Carretilla 7	0	2	3	0	

Imagen 37. Experimento 6. Rutas.

Después de varias ejecuciones para encontrar el número de carretillas idílico para satisfacer la demanda, se encuentra que éste es siete y tiene un coste total de 4.475,00€. Este coste se ve disminuido respecto al experimento anterior pues en este caso sí se está sirviendo a toda la demanda, mientras que en el experimento anterior no. Como se ha mencionado en otros apartados del trabajo, el no servir a la demanda a tiempo repercute en un gran coste para la empresa dado que su imagen se ve dañada.

Experimento 7: L=4, J=4, R=10

Con este experimento se tratará de demostrar que existe un máximo de carretillas que se utilizarían en el proceso, es decir, se pueden disponer de 20 carretillas pero en realidad, para satisfacer la demanda con el menor coste posible, el contar sólo con la mitad de ellas sería suficiente.

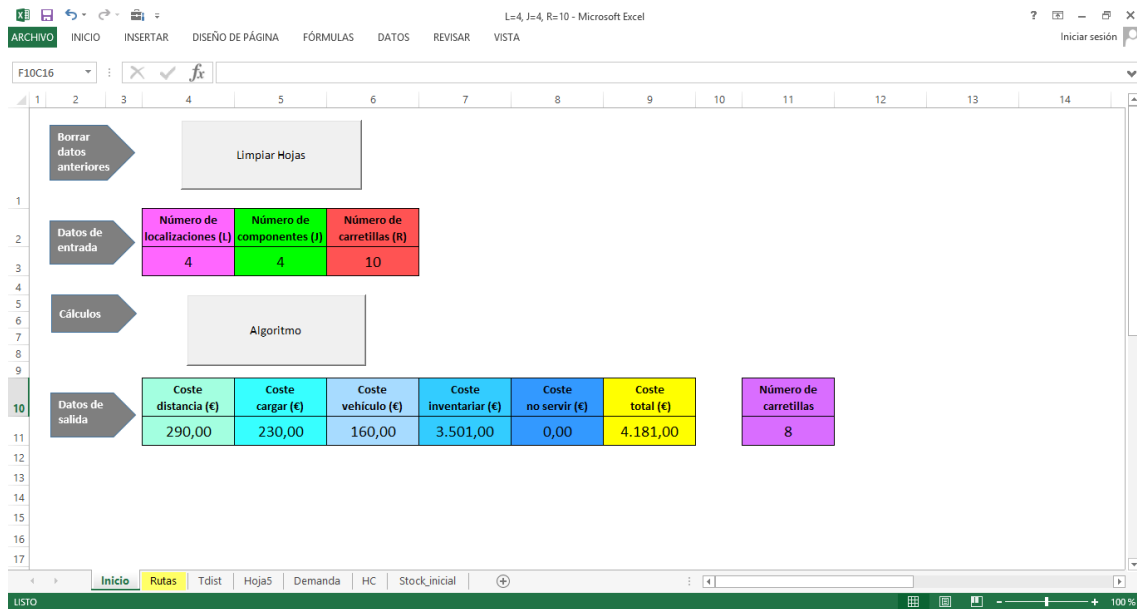


Imagen 38. Experimento 7. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	1	0
Carretilla 2	0	1	2
Carretilla 3	0	2	3
Carretilla 4	0	3	1
Carretilla 5	0	1	0
Carretilla 6	0	1	2
Carretilla 7	0	2	3
Carretilla 8	0	3	0

Imagen 39. Experimento 7. Rutas.

Como se ha mencionado antes de realizar el experimento, es cierto que aunque la empresa contara con diez carretillas, con ocho bastaría para poder realizar correctamente el suministro. Estas dos carretillas sobrantes podrían revenderse para así obtener un beneficio ya que si no, estarían paradas y esto no supone ningún beneficio para la empresa.

Experimento II: problemas de grandes dimensiones

A partir de aquí, como se va experimentar con situaciones de mayor número de localizaciones y de componentes distintos, se va a necesitar nuevos datos de: la demanda, la distancia entre una localización y otra, el coste por componente y el stock inicial, además del cambio de valores de dos variables. Para simplificar los datos, se ha utilizado la siguiente información:

- Demanda: seguirá habiendo una necesidad cada 5 unidades de tiempo y la demanda se repetirá cada 8 unidades.
- Distancia recorrida: todas las distancias entre localizaciones será de 10 unidades. La distancia entre el warehouse y la localización será de: 2,3,4,5,6 recurrente.
- Coste por componente: 6,7 recurrente.
- Stock inicial: todos serán de 8u.
- Cantidad máxima a transportar por cada carretilla: 50 unidades.
- Tiempo máximo: 50 unidades de tiempo.

Además de todo esto, hay que tener en cuenta que el algoritmo utilizado busca mejores soluciones de forma aleatoria y como el programa no se ejecuta tantas veces como rutas posibles existen, en cada ejecución se podrían encontrar resultados distintos.

Experimento 8: L=10, J=4, R=2

En la primera situación se aumentará el número de localizaciones hasta 10 y se mantendrá el número de componentes (J=4).

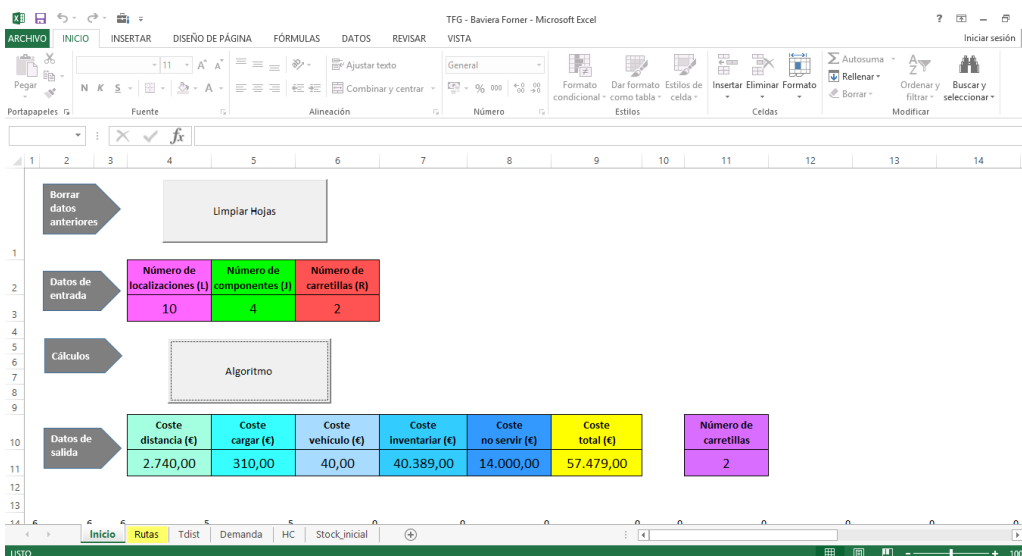


Imagen 40. Experimento 8. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	2	9	0	8	4	3	5	0	7	2	0	1	8	0	5	6	0
Carretilla 2	0	1	8	0	5	6	0	9	1	0	4	3	5	0	7	0		

Imagen 41. Experimento 8. Rutas.

Una vez aumentado el número de localizaciones, se observa como teniendo únicamente dos carretillas no se puede satisfacer a toda la demanda. Esto resulta lógico ya que el carretillero deberá desplazarse a más localizaciones por ruta de las que se desplazaba antes y con lo cual, puede que no tenga tiempo suficiente para recorrerlas todas en el momento en el que lo requieren.

Experimento 9: L=10, J=4, R=10

Como en el experimento anterior no se ha satisfecho la demanda de los diferentes puestos de trabajo, en este experimento se aumentará el número de carretillas para ver si así se consigue satisfacer ésta.

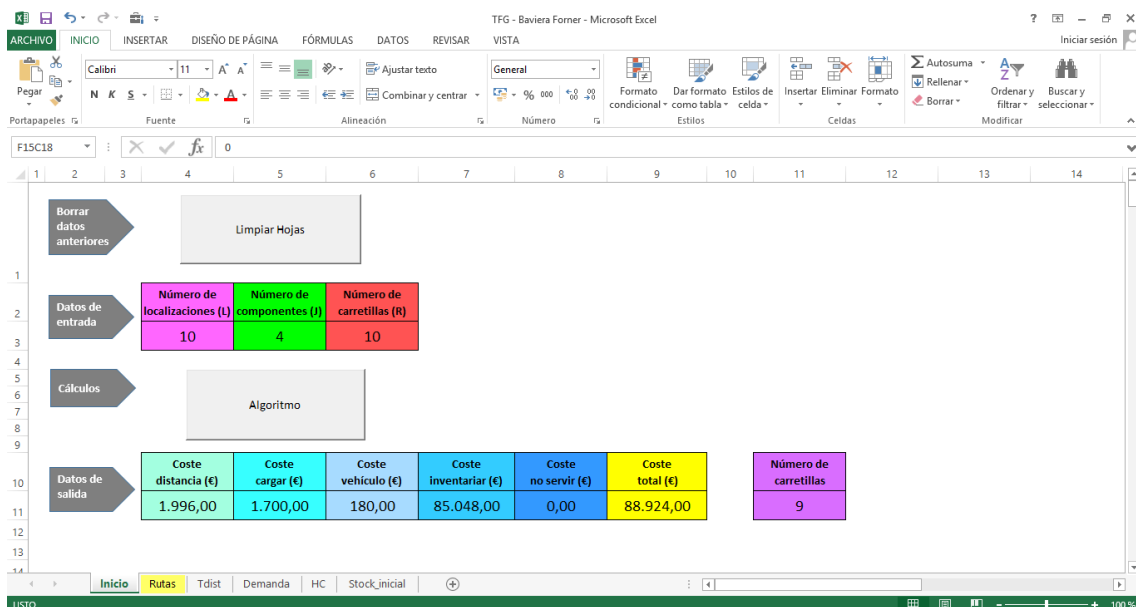


Imagen 42. Experimento 9. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	3	2	4	9	0
Carretilla 2	0	9	7	0		
Carretilla 3	0	8	1	0		
Carretilla 4	0	1	6	5	0	
Carretilla 5	0	5	3	2	4	0
Carretilla 6	0	9	7	0		
Carretilla 7	0	7	8	0		
Carretilla 8	0	1	6	0		
Carretilla 9	0	6	5	0		

Imagen 43. Experimento 9. Rutas.

En esta ocasión, para satisfacer la demanda de diez localizaciones con cuatro componentes distintos, se ha tenido que utilizar nueve carretillas. Aunque en la pantalla se muestren dos valores distintos en cuanto al número de carretillas, esto se debe a que en los datos de entrada se introduce el número de carretillas que se tiene y el que se debería utilizar para suministrar los diferentes puestos, pero, como este programa también reduce el número de carretillas a utilizar, el programa ha decidido que sería mejor hacer uso de nueve de las diez carretillas disponibles. Esto se realiza de esta manera puesto que el uso de una carretilla menos equivale a un coste menor por vehículo.

Experimento 10: L=10, J=10, R=10

En esta ejecución se aumentará el número de componentes distintos que se pueden encontrar en cada localización.

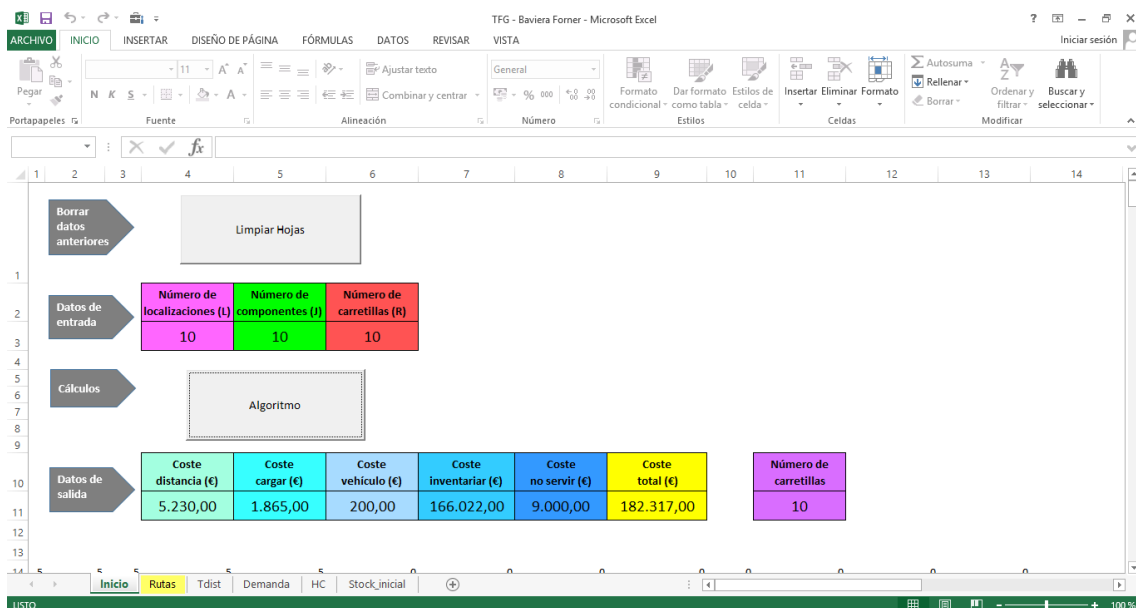


Imagen 44. Experimento 10. Inico.

Localizaciones (L=0=warehouse)

Carretilla 1	0	5	0	3	1	0	7	8	0
Carretilla 2	0	5	6	0	1	5	0	8	0
Carretilla 3	0	6	0	5	0	8	3	0	
Carretilla 4	0	6	2	0	5	6	0	3	1
Carretilla 5	0	2	4	0	6	0	1	0	
Carretilla 6	0	9	0	6	2	0			
Carretilla 7	0	9	7	0	2	4	0		
Carretilla 8	0	7	0	4	9	0			
Carretilla 9	0	8	0	9	7	0			
Carretilla 10	0	8	3	0	7	0			

Imagen 45. Experimento 10. Rutas.

En este experimento no se sirve toda la demanda a tiempo, con lo que se deberá de aumentar el número de carretillas para ver cuántas se necesitan para llegar a tiempo de suministrar todas las localizaciones.

Además, mencionar que entre el experimento anterior y éste se ve una gran diferencia en cuanto a los costes debido a que ahora se introducen 6 componentes más por puesto de trabajo, con lo cual, en esta ocasión se inventaría un mayor número de unidades, lo que significa que el coste de inventariar aumenta considerablemente.

Experimento 11: L=10, J=10, R=50

Como en el experimento anterior no se ha llegado a servir a tiempo toda la demanda, en esta ejecución se aumentará el número de carretillas.

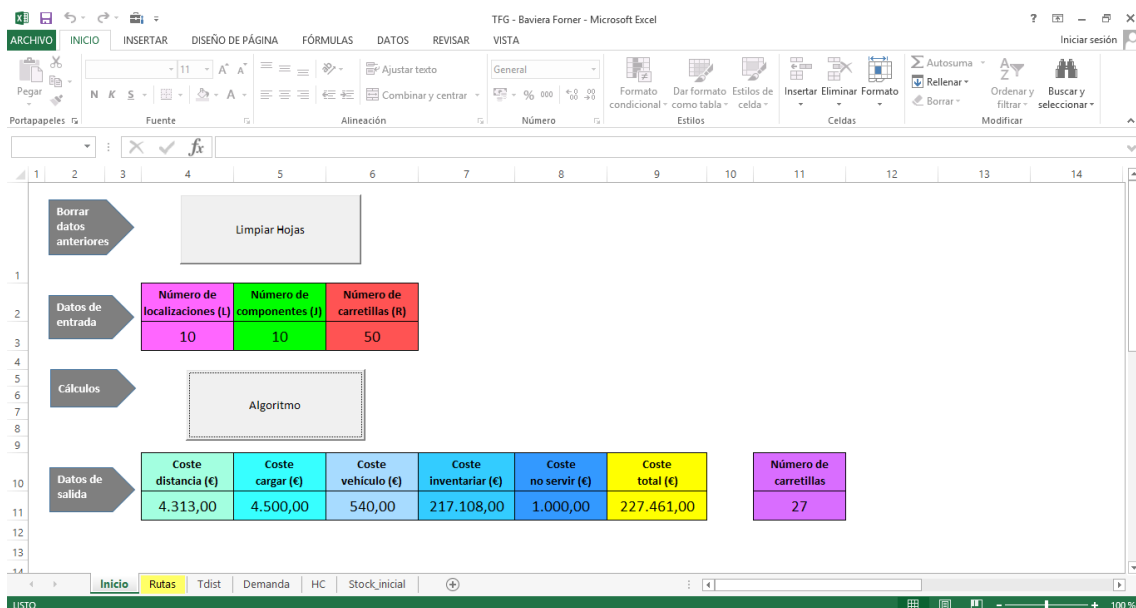


Imagen 46. Experimento 11. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	2	5	0
Carretilla 2	0	5	0	0
Carretilla 3	0	5	4	0
Carretilla 4	0	4	0	0
Carretilla 5	0	4	1	6
Carretilla 6	0	6	0	0
Carretilla 7	0	6	3	0
Carretilla 8	0	3	0	0
Carretilla 9	0	3	8	0
Carretilla 10	0	8	9	0
Carretilla 11	0	9	0	0
Carretilla 12	0	9	7	0
Carretilla 13	0	7	0	0
Carretilla 14	0	7	2	0
Carretilla 15	0	5	0	0
Carretilla 16	0	5	4	0
Carretilla 17	0	4	0	3
Carretilla 18	0	4	1	0
Carretilla 19	0	6	0	0
Carretilla 20	0	6	3	0
Carretilla 21	0	3	0	0
Carretilla 22	0	3	8	0
Carretilla 23	0	8	0	0
Carretilla 24	0	9	0	0
Carretilla 25	0	9	7	0
Carretilla 26	0	7	0	0
Carretilla 27	0	7	0	0

Imagen 47. Experimento 11. Rutas.

En este caso, el número de carretillas se aumenta hasta 50 pero realmente, con 27 tendría suficiente. En realidad, como se puede observar en la imagen superior esta vez pese a tener el número de carretillas suficiente, no se llega a servir a toda la demanda. Esto es porque en una de las rutas que hace, aunque la carretilla sí tiene capacidad suficiente como para transportar el material de una localización a otra, ésta no llega a tiempo debido a las distancias que recorre. Como las distancias no se pueden modificar, habrá que aumentar el stock. Evidentemente el stock inicial tampoco se puede alterar conforme el usuario quiera, pero viendo esto se llega a la conclusión de que el stock que se debería de modificarse debería ser el stock de seguridad. Éste debería ser mayor, ya que en caso contrario, existirían momentos en los que la demanda no sería servida a tiempo.

Experimento 12: L=10, J=10, R=27

Para demostrar lo anteriormente mencionado, se cambiará el stock inicial de todos los puestos de trabajo en una unidad. Esto equivaldría a tener una unidad de stock de seguridad.

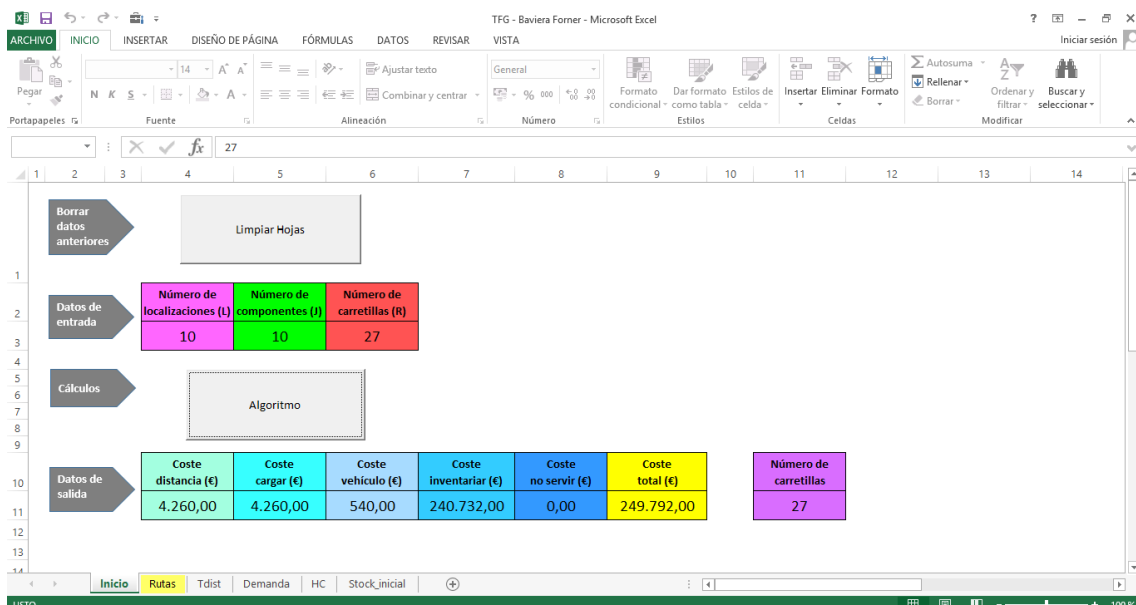


Imagen 48. Experimento 12. Inicio

Localizaciones (L=0=warehouse)

Carretilla 1	0	9	0	2	4	0
Carretilla 2	0	9	0			
Carretilla 3	0	1	0			
Carretilla 4	0	1	8	0		
Carretilla 5	0	8	0			
Carretilla 6	0	8	3	0		
Carretilla 7	0	3	0			
Carretilla 8	0	6	0			
Carretilla 9	0	6	5	0		
Carretilla 10	0	5	0			
Carretilla 11	0	5	7	0		
Carretilla 12	0	7	0			
Carretilla 13	0	2	0			
Carretilla 14	0	2	4	0		
Carretilla 15	0	9	0			
Carretilla 16	0	9	0	0		
Carretilla 17	0	1	0			
Carretilla 18	0	1	8	0		
Carretilla 19	0	8	0			
Carretilla 20	0	8	3	0		
Carretilla 21	0	3	0	0		
Carretilla 22	0	6	0			
Carretilla 23	0	6	5	0		
Carretilla 24	0	5	0			
Carretilla 25	0	5	7	0		
Carretilla 26	0	7	0			
Carretilla 27	0	2	0			

Imagen 49. Experimento 12. Rutas.

El aumentar el stock inicial en una unidad ha permitido que se suministre a tiempo a toda la demanda.

Experimento 13: L=30, J=20, R=15

Para realizar esta simulación se mantendrá el stock de seguridad de una unidad y mediante éste intentará satisfacer toda la demanda

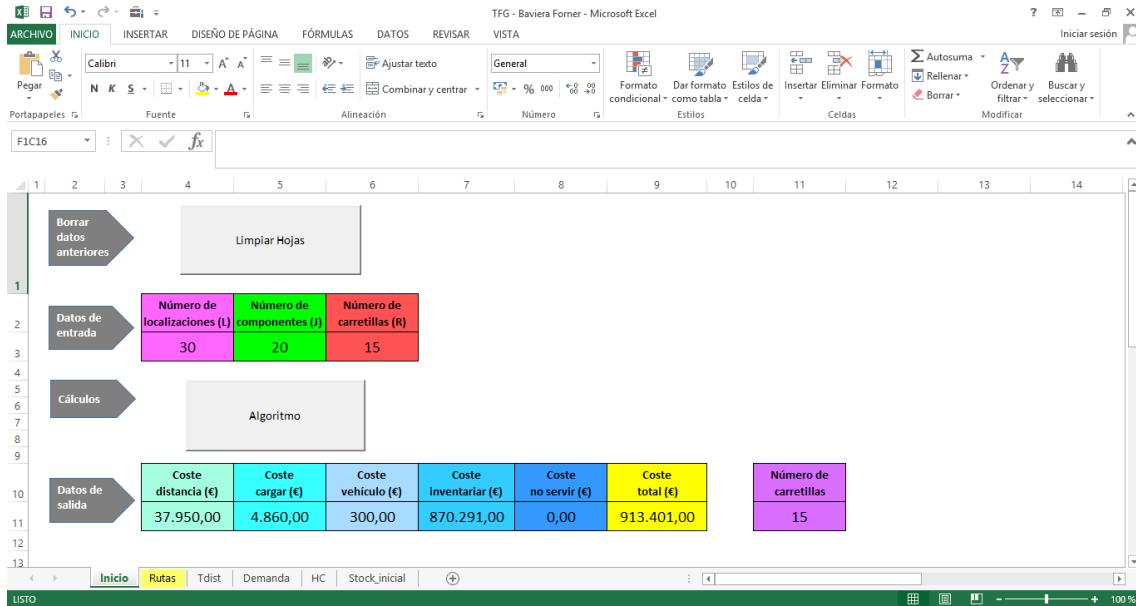


Imagen 50. Experimento 13. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	8	0	9	7	0	21	29	0	25	14	0	1	3	0	2	0	18	0	7	0	29	0	25	0	1	0	26	0	27	0	
Carretilla 2	0	8	0	7	0	29	0	14	0	3	23	0	2	0	18	0	7	13	0	29	0	25	14	0	1	0	26	0				
Carretilla 3	0	8	0	7	0	29	0	14	0	23	0	24	0	18	0	13	0	29	0	14	0	1	0	26	2	0						
Carretilla 4	0	8	18	0	7	13	0	29	22	0	14	16	0	23	0	24	0	18	10	0	13	0	29	22	0	14	0	1	3	0	2	0
Carretilla 5	0	18	0	13	0	22	0	16	0	23	6	0	24	0	10	0	13	0	22	0	14	0	3	23	0	2	0					
Carretilla 6	0	18	0	13	0	22	0	16	0	6	0	15	0	10	0	4	0	22	0	16	0	23	0	2	0							
Carretilla 7	0	18	10	0	13	4	0	22	12	0	16	17	0	6	0	15	0	10	11	0	4	0	22	12	0	16	0	23	0	24	0	
Carretilla 8	0	10	0	4	0	12	0	17	0	6	19	0	15	0	11	0	4	0	12	0	16	0	23	6	0	24	0					
Carretilla 9	0	10	0	4	0	12	0	17	0	19	0	27	0	11	0	4	28	0	12	0	16	17	0	6	0	24	0					
Carretilla 10	0	10	11	0	4	28	0	12	5	0	17	20	0	19	0	27	0	11	0	28	0	12	0	17	0	6	0	24	15	0		
Carretilla 11	0	11	0	28	0	5	0	20	0	19	26	0	27	0	9	0	28	0	5	0	17	0	6	0	15	0						
Carretilla 12	0	11	0	28	0	5	0	20	0	26	0	8	0	9	0	28	21	0	5	0	17	20	0	19	0	15	0					
Carretilla 13	0	11	9	0	28	21	0	5	25	0	20	1	0	26	0	8	0	9	0	21	0	5	0	20	0	19	0	15	27	0		
Carretilla 14	0	9	0	21	0	25	0	1	0	26	2	0	8	0	9	7	0	21	0	5	25	0	20	0	19	0	27	0				
Carretilla 15	0	9	0	21	0	25	0	1	0	2	0	8	0	7	0	21	0	25	0	20	0	19	26	0	27	0						

Imagen 51. Experimento 13. Rutas.

En este caso, la demanda sí queda satisfecha.

Experimento 14: L=50, J=20, R=20

Este último experimento mostrará como el programa funciona para 50 localizaciones y 20 componentes distintos.

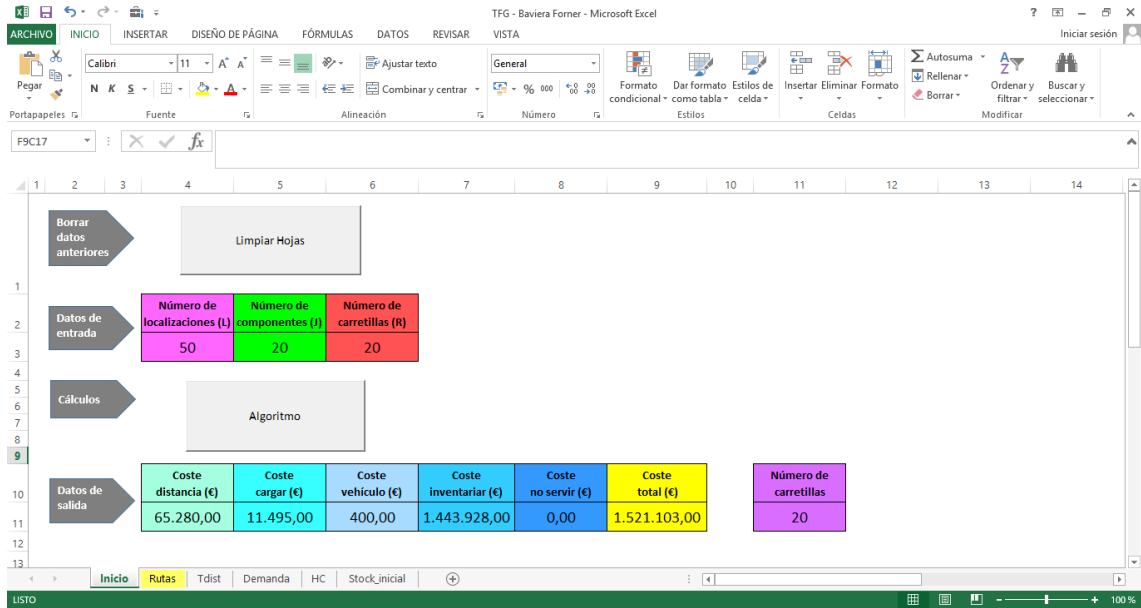


Imagen 52. Experimento 14. Inicio.

Localizaciones (L=0=warehouse)

Carretilla 1	0	34	0	18	0	3	0	48	0	16	0	35	25	0	15	28	0	26	0	11	0	30	0	1	0	12	0	40	0	38	0	28	0	26	0									
Carretilla 2	0	34	0	18	31	0	3	0	48	24	0	16	0	25	0	28	0	26	0	11	0	30	0	1	0	12	0	40	0	38	0	28	0	26	0									
Carretilla 3	0	34	0	31	0	3	4	0	24	0	16	43	0	25	0	28	0	26	6	0	11	23	0	30	47	0	1	0	12	0	40	0	38	0	28	0	26	0						
Carretilla 4	0	34	7	0	31	0	4	42	0	24	0	43	0	38	0	28	22	0	6	0	23	0	47	0	1	37	0	12	29	0	40	20	0	38	27	0	28	22	0	26	6	0		
Carretilla 5	0	7	0	31	13	0	42	0	24	32	0	43	0	38	0	22	0	6	0	23	0	47	0	37	0	29	0	20	0	27	0	22	0	6	0	0	0	0	0	0				
Carretilla 6	0	7	0	13	0	42	0	32	0	43	40	0	38	0	22	0	6	34	0	23	0	47	0	37	0	29	0	20	0	27	0	22	0	6	0	0	0	0	0	0				
Carretilla 7	0	7	36	0	13	0	42	3	0	32	0	40	0	27	0	22	2	0	34	0	23	18	0	47	8	0	27	33	0	29	49	0	20	44	0	27	19	0	22	2	0	6	0	
Carretilla 8	0	36	0	13	21	0	5	0	32	12	0	40	0	27	0	2	0	34	0	18	0	8	0	33	0	49	0	44	0	19	0	2	0	0	0	0	0	0	0	0	0			
Carretilla 9	0	36	0	21	0	5	0	12	0	40	20	0	27	0	2	0	34	0	18	0	8	0	33	0	49	0	44	0	19	0	2	0	0	0	0	0	0	0	0	0	0			
Carretilla 10	0	36	9	0	21	0	5	1	0	12	0	20	0	19	0	2	0	34	7	0	18	31	0	8	3	0	33	0	49	0	44	0	19	0	2	0	0	0	0	0	0	0		
Carretilla 11	0	9	0	21	30	0	1	0	12	29	0	20	0	19	0	2	10	0	7	0	31	0	3	0	48	0	17	0	46	0	39	0	10	0	0	0	0	0	0	0	0	0		
Carretilla 12	0	9	0	30	0	1	0	29	0	20	44	0	19	0	10	0	7	0	31	0	3	0	48	0	17	0	46	0	39	0	10	0	0	0	0	0	0	0	0	0	0	0		
Carretilla 13	0	9	11	0	30	0	1	37	0	29	0	44	0	39	0	10	0	7	36	0	31	13	0	3	4	0	48	0	17	0	46	0	39	0	10	0	0	0	0	0	0	0	0	
Carretilla 14	0	11	0	47	0	37	0	29	49	0	44	0	39	0	10	41	0	36	0	13	0	4	42	0	48	24	0	17	16	0	46	35	0	39	45	0	10	41	0	0	0	0	0	
Carretilla 15	0	11	0	47	0	37	0	49	0	44	46	0	39	45	0	41	0	36	0	13	0	42	0	24	0	16	0	35	0	45	0	41	0	0	0	0	0	0	0	0	0	0		
Carretilla 16	0	11	23	0	47	0	37	33	0	49	0	46	0	45	0	41	0	36	0	13	0	42	0	24	0	16	0	35	0	45	0	41	0	0	0	0	0	0	0	0	0	0	0	
Carretilla 17	0	23	0	8	0	33	0	49	17	0	46	0	45	0	41	14	0	36	9	0	13	21	0	42	5	0	24	32	0	16	43	0	35	25	0	45	15	0	41	14	0	0	0	
Carretilla 18	0	23	0	8	0	33	0	17	0	46	35	0	45	15	0	14	0	9	0	21	0	5	0	32	0	43	0	25	0	15	0	14	0	0	0	0	0	0	0	0	0	0	0	
Carretilla 19	0	23	18	0	8	0	33	48	0	17	0	35	0	15	0	14	0	9	0	21	0	5	0	32	0	43	0	25	0	15	0	14	0	0	0	0	0	0	0	0	0	0	0	
Carretilla 20	0	18	0	3	0	48	0	17	16	0	35	0	15	0	14	26	0	9	11	0	21	30	0	5	0	32	0	43	0	25	0	15	0	14	0	0	0	0	0	0	0	0	0	0

Imagen 53. Experimento 14. Rutas

Mediante este resultado se puede ver como sí es posible satisfacer la demanda con 50 localizaciones y 20 componentes.

6. CONCLUSIONES FINALES Y RECOMENDACIONES

Mediante este trabajo se ha podido ver que siempre existe alguna manera de conseguir que una empresa reduzca los costes. Puede que no sea fácil detectar esas zonas donde poder actuar, pero siempre hay que estudiar cada situación para poder ver cómo mejorarla.

Aunque Ford sea una empresa que lleve activa mucho tiempo y cuente con personal cualificado que siempre intenta mejorar la situación de la misma, ya sea mediante reducciones de coste o de tiempos, siempre queda algo que explorar. En este caso, al visitar la planta de Ford se vio que un punto donde se podría lograr reducir los costes era a la hora de suministrar cada línea con sus correspondientes piezas. Para poder reducir estos costes, se diseñó un modelo matemático y una herramienta informática.

Una vez realizadas y ejecutadas ambas herramientas (el modelo matemático y el programa informático), se puede deducir que ambas son muy útiles, pero dependiendo de qué se esté buscando y de cómo de compleja sea la situación, se deberá resolver el problema de una manera u otra. Si la situación no es muy compleja y se cuenta con tiempo para resolverlo, un modelo matemático sería perfecto ya que puede encontrar una solución óptima, mientras que el algoritmo puede que simplemente encuentre una buena solución, pero no la óptima. Por otro lado, si la situación fuera compleja, habría que utilizar un algoritmo ya que éste es más rápido a la hora de resolver y da buenas soluciones, aunque éstas no sean siempre las óptimas.

Añadiendo a lo anterior, hay que tener en cuenta que a medida que haya más localizaciones y componentes, las carretillas deberán contar con una mayor capacidad, ya que en caso contrario tendrán que realizar más viajes para poder servir a tiempo todos los puestos. Si no hubiese presupuesto suficiente para comprar carretillas con mayores capacidades, se debería aumentar el stock de seguridad. Durante el proyecto se había asumido que el stock de seguridad era cero, aunque mediante la experimentación se vio que si no se contaba con un stock de seguridad, la resolución del problema nunca era buena, pues había situaciones en las que el material se requería antes de que éste llegase. Mediante el stock de seguridad, esto se hubiese evitado. El nivel de stock de seguridad que debería de haber tendría que ser suficiente como para cubrir estos posibles fallos, pero no demasiado alto dado que contribuiría al aumento de costes.

Evidentemente, la utilización práctica de estas dos herramientas creadas proporciona la posibilidad de cargar el máximo número de piezas por carretilla para así realizar un menor número de viajes y además, tratando de servir cada puesto en el momento justo en el que lo precise. Todo esto contribuye a una disminución de los costes respecto a la situación inicial.

Este TFG ha intentado representar la situación inicial de la manera más real posible. Para conseguirlo, además de la citada visita a Ford, se ha extraído información de un operario del área de Prensas, un operario de la línea de montaje del área Trim, un responsable de mantenimiento del área de Prensas y un ingeniero, miembro del Equipo Directivo.

A nivel personal, la realización de este TFG me ha permitido poder aumentar mis conocimientos en el área de programación. Durante la carrera, en las asignaturas de “Métodos Cuantitativos” y “PCPO” se pudo ver un poco de programación, pero realmente no con tanta profundidad como durante el TFG.

Añadiendo a lo anterior, mientras realizaba las prácticas en una empresa del sector de la automoción, he tenido la oportunidad de poner en práctica estos nuevos conocimientos de programación con Visual Basic y me ha resultado muy provechoso.

Este TFG también me ha ayudado a comprender mejor como funciona un operador logístico de grandes dimensiones tal y como es Ford. Además, ha sido de gran interés realizar el TFG allí puesto que estoy focalizada en el sector de la automoción. Es de interés este sector ya que es un sector cambiante, en el que hay muchos competidores y siempre se tiene que estar a la última, sin permitirse fallos ya que si no, la competencia podría aprovecharse de ello y aventajarnos.

7. BIBLIOGRAFÍA

ARTURO DÍAZ PÉREZ, Análisis y Diseño de Algoritmos, 2014.

BARBARA C RICHARDSON, KENT B. JOSCELYN et Al., Limitations on the use of mathematical models in transportation policy analysis, 1979.

FREDERIC S. HILLIER, Introducción a la investigación de operaciones, 1997.

GABRIELE KAISER, WERNER BLUM et Al., Trends in teaching and learning of mathematical modelling, 2011.

H. PAUL WILLIAMS, Model Building in Mathematical Programming, 2013.

N. PAROLINI AND A. QUARTERONI, Mathematical models and numerical simulations for the America's Cup, Comp. Meth. Appl. Mech. Eng.,173(2005),1001–1026.

RLEC. REVERSE LOGISTICS EXECUTIVES' COUNCIL, The Council of Logistics Management, 2009.

8. ANEXOS DE LA MEMORIA

8.1 Modelo matemático

TITLE TFG;

DATA

R:=2;

L:=10;

J:=4;

T:=10;

M:=100000;

CAP=10;

MC=5;

A=20;

TC=10;

INDEX

r:=1..R;

l:=0..L;

l1:=0..L;

l2:=0..L;

l3:=0..L;

l4:=0..L;

j:=1..J;

t:=1..T;

DATA

Tdist[l1,l2]:=EXCELRange("TFG.xls","Tdist");

dem[t,j,l>0] :=EXCELRange("TFG.xls","dem");

ST0[j,l>0] :=EXCELRange("TFG.xls","ST0");

HC[j]:=EXCELRange("TFG.xls","HC");

INTEGER VARIABLES

tloc[r,l];

f[j,l1,l2,r];

c[j,l,r,t];

stock[j,t,l1];

BINARY VARIABLES

x[r,l1,l2];

w[r,l1];

a[r];

MODEL MIN

sum(r,l1,l2:TC*Tdist*x)+MC*sum(j,l1,l2,r:f)+A*sum(r:a)+sum(j,t,l1:HC*stock);

SUBJECT TO

C1_[l2>0]: sum(r,l1<>l2:x)=1;

C2_[l1>0]: sum(r,l2<>l1:x)=1;

C3_[l1>0]: sum(r:w)=1;

C4_[r,l1>0]: w=sum(l2<>l1:x);

C5_ $[r,l4]$: $\text{sum}(l3:x[r,l1:=l3,l2:=l4] \text{ where } l3<>l4)=\text{sum}(l3:x[r,l1:=l4,l2:=l3] \text{ where } l3<>l4)$;
 C6_ $[l2=0]$: $\text{sum}(r,l1>0:x[r,l1,l2])=\text{sum}(r:a)$;
 C7_ $[l1=0]$: $\text{sum}(r,l2>0:x[r,l1,l2])=\text{sum}(r:a)$;
 C8_ $[r]$: $\text{sum}(l1,l2<>l1:x)\leq M*a$;
 C9_ $[r]$: $a>=a[r+1]$;
 C10_: $\text{sum}(r:a)\leq R$;
 C11_ $[r,l3<>0]$: $\text{tloc}[r,l:=l3]>=\text{Tdist}[l1:=0,l2:=l3]-M*(1-x[r,l1:=0,l2:=l3])-M*(2-w[r,l1:=0]-w[r,l1:=l3])$;
 C12_ $[r,l3,l4<>l3]$: $\text{tloc}[r,l:=l4]>=\text{tloc}[r,l:=l3]+\text{Tdist}[l1:=l3,l2:=l4]-M*(1-x[r,l1:=l3,l2:=l4])-M*(2-w[r,l1:=l3]-w[r,l1:=l4])$;
 C13_ $[l1,r]$: $\text{sum}(j,l2>0:f[j,l1=0,l2,r])=\text{CAP}$;
 C14_ $[j,l,l3<>l,l4<>l,r,t]$: $f[j,l1:=l3,l2:=l4,r]+c[j,l:=l3,r,t]>=f[j,l1:=l,l2:=l3,r]-M*(1-x[r,l1:=l,l2:=l3])-M*(1-x[r,l1:=l3,l2:=l4])-M*(2-w[r,l1:=l]-w[r,l1:=l3]-w[r,l1:=l4])$;
 C15_ $[j,l1>0]$: $\text{stock}[j,t:=1,l1]=\text{STO}[j,l:=l1]-\text{dem}[1,j,l:=l1]+\text{sum}(r:c[j,l:=l1,r,1])$;
 C16_ $[j,l1>0,t>1]$: $\text{stock}[j,t,l1]=\text{stock}[j,t-1,l1]-\text{dem}[t,j,l:=l1]+\text{sum}(r:c[j,l:=l1,r,t])$;
 C17_ $[j,l1,t>1]$: $\text{stock}>=\text{dem}[t,j,l:=l1]$;

END

8.2 Herramienta informática

Private Sub CommandButton21_Click()

Dim a As Single
 Dim b As Integer
 Dim c As Integer
 Dim d As Integer
 Dim f As Integer
 Dim w As Integer
 Dim x As Single
 Dim y As Single

Dim R As Integer
 Dim L As Integer
 Dim J As Integer
 Dim tiempo As Integer
 Dim cantmax As Integer
 Dim M As Integer
 Dim MC As Integer
 Dim TC As Integer
 Dim coste_vehiculo As Integer
 Dim HC() As Integer
 Dim tdist() As Integer
 Dim stock_inicial() As Single
 Dim transporte() As Integer

Dim ruta() As Integer
 Dim aleatorio As Integer
 Dim pos_aleat() As Integer
 Dim acum_tiempo() As Integer
 Dim cantidad() As Integer
 Dim contador As Single
 Dim intermedio() As Integer
 Dim cargar() As Integer
 Dim posicion() As Integer
 Dim num_columnas As Single
 Dim columnas() As Single

Dim multiplicador As Integer
 Dim auxiliar1 As Single
 Dim redondeo As Single

Dim vector_intermedio() As Single
 Dim vector_cargar() As Single
 Dim vector_posicion() As Single
 Dim nuevo_intermedio() As Single
 Dim nuevo_cargar() As Single
 Dim nuevo_posicion() As Single

Dim mejor_matriz() As Single
 Dim mejor_cargar() As Single
 Dim mejor_posicion() As Single
 Dim guardar_aleat() As Integer
 Dim nuevo() As Integer
 Dim suma_tiempo() As Integer
 Dim min_tiempo() As Integer

Dim tiempo_mio() As Integer
 Dim llegada() As Single
 Dim llegada_acum() As Single

Dim coste_dist As Single
 Dim total_cargar As Single
 Dim coste_inv As Single
 Dim demanda_sin_servir As Single
 Dim coste_total As Single

Dim sol_ruta() As Single
 Dim sol_coste_dist As Single
 Dim sol_total_cargar As Single
 Dim sol_coste_inv As Single
 Dim sol_contador As Single
 Dim sol_demanda_sin_servir As Single
 Dim sol_num_columnas As Single
 Dim sol_coste_total As Single
 Dim sol_multiplicador As Integer
 Dim sol_mejor_matriz() As Single

'Datos'

```
L = Worksheets("Inicio").Cells(3, 4).Value
J = Worksheets("Inicio").Cells(3, 5).Value
R = Worksheets("Inicio").Cells(3, 6).Value
tiempo = 50
cantmax = 50
TC = 10
MC = 5
ReDim HC(J)
For x = 1 To J
    HC(x) = Worksheets("HC").Cells(2, x).Value
Next x
coste_vehiculo = 20
M = 500
```

'Redimensionar los vectores y matrices'

```
ReDim ordenado(L - 1)
ReDim ruta(L - 1)
ReDim pos_aleat(L - 1)
ReDim transporte(tiempo, J * (L - 1))
ReDim cantidad(M * 6)
ReDim tdist(L, L)
ReDim stock_inicial(J * (L - 1))
ReDim acum_tiempo(M * 6)
ReDim cargar(M * 6, M)
ReDim intermedio(M * 6, M)
ReDim posicion(M * 6, M)
```

'Leer los datos de entrada'

```
For x = 1 To tiempo
    For y = 1 To J * (L - 1)
        transporte(x, y) = Worksheets("Demanda").Cells(2 + x, 1 + y).Value
    Next y
Next x

For x = 1 To L
    For y = 1 To L
        tdist(x, y) = Worksheets("Tdist").Cells(1 + x, 1 + y).Value
    Next y
Next x

For x = 1 To J * (L - 1)
    stock_inicial(x) = Worksheets("Stock_inicial").Cells(3, x).Value
Next x
```

'Inicializar las variables'

```
ReDim sol_ruta(L - 1)
For x = 1 To L - 1
    sol_ruta(x) = 0
Next x

sol_coste_dist = 0
sol_total_cargar = 0
sol_contador = 0
sol_coste_inv = 0
sol_demanda_sin_servir = 0
sol_coste_total = 30000000
```

'Ejecutar el programa 100 veces y que se quede con la mejor solución encontrada (la de menor coste)'

```
For w = 1 To 100
```

'Crear una ruta aleatoria provisional'

```
For x = 1 To L - 1
    ruta(x) = 0
Next x

For x = 1 To L - 1
    pos_aleat(x) = 0
Next x

a = 1
For x = 1 To 6 * M
    If a <= L - 1 Then
        aleatorio = Int(((L - 1) * Rnd()) + 1)
        If pos_aleat(aleatorio) <> aleatorio Then
            ruta(a) = aleatorio
            a = a + 1
            pos_aleat(aleatorio) = aleatorio
        End If
    End If
Next x
```

'Cantidad a transportar en cada viaje, sabiendo que hay un máximo de capacidad por cada caarretilla'

```
For x = 1 To M * 6
    cantidad(x) = 0
    acum_tiempo(x) = 0
Next x
```

```

For x = 1 To M * 6
  For y = 1 To M
    intermedio(x, y) = 0
    cargar(x, y) = 0
    posicion(x, y) = 0
  Next y
Next x

a = 0
f = 1
For x = 1 To tiempo
  For y = 1 To L - 1
    For b = 1 To J
      If transporte(x, -J + J * ruta(y) + b) > 0 Then
        If cantidad(a) + transporte(x, -J + J * ruta(y) + b) > cantmax Or a = 0 Or
acum_tiempo(a) + tdist(ruta(y - 1) + 1, ruta(y) + 1) > tiempo Then
          a = a + 1
          acum_tiempo(a) = tdist(1, ruta(y) + 1)
          f = 1
        Else
          acum_tiempo(a) = acum_tiempo(a) + tdist(ruta(y - 1) + 1, ruta(y) + 1)
        End If
        intermedio(a, f) = ruta(y)
        cargar(a, f) = transporte(x, -J + J * ruta(y) + b)
        posicion(a, f) = -J + J * ruta(y) + b
        cantidad(a) = cantidad(a) + transporte(x, -J + J * ruta(y) + b)
        f = f + 1
      End If
    Next b
  Next y
Next x

```

'para saber cuántos R ("contador") serían necesarios'

```

contador = 0
For x = 1 To M * 6
  If cantidad(x) <> 0 Then
    contador = contador + 1
  End If
Next x

ReDim columnas(contador)

For x = 1 To contador
  columnas(x) = 0
  For y = 1 To M
    If intermedio(x, y) <> 0 Then
      columnas(x) = columnas(x) + 1
    End If
  Next y
Next x

```

```
num_columnas = 0
For x = 1 To contador
  If num_columnas < columnas(x) Then
    num_columnas = columnas(x)
  End If
Next x
```

'Multiplicador'

```
redondeo = contador / R
auxiliar1 = redondeo - Int(redondeo)
If auxiliar1 = 0 Then
  multiplicador = Int(redondeo)
Else
  multiplicador = Int(redondeo) + 1
End If
```

'Que sólo haya tantas rutas como R'

```
ReDim nuevo_intermedio(contador, multiplicador * num_columnas)
ReDim nuevo_cargar(contador, multiplicador * num_columnas)
ReDim nuevo_posicion(contador, multiplicador * num_columnas)
ReDim mejor_matriz(contador, multiplicador * num_columnas)
ReDim mejor_cargar(contador, multiplicador * num_columnas)
ReDim mejor_posicion(contador, multiplicador * num_columnas)
ReDim suma_tiempo(contador, multiplicador)
ReDim min_tiempo(contador, multiplicador)
ReDim guardar_aleat(multiplicador * num_columnas)
ReDim nuevo(multiplicador * num_columnas)
```

```
For x = 1 To contador
  For y = 1 To multiplicador * num_columnas
    nuevo_intermedio(x, y) = 0
    nuevo_cargar(x, y) = 0
    nuevo_posicion(x, y) = 0
    mejor_matriz(x, y) = 0
    mejor_cargar(x, y) = 0
    mejor_posicion(x, y) = 0
  Next y
Next x
```

```
For x = 1 To contador
  For y = 1 To multiplicador
    min_tiempo(x, y) = 0
  Next y
Next x
```

```
If contador > R Then

ReDim vector_intermedio(contador * num_columnas)
ReDim vector_cargar(contador * num_columnas)
ReDim vector_posicion(contador * num_columnas)

For x = 1 To contador * num_columnas
    vector_intermedio(x) = 0
    vector_cargar(x) = 0
    vector_posicion(x) = 0
Next x

For x = 1 To contador
    For y = 1 To num_columnas
        vector_intermedio(y + (x - 1) * num_columnas) = intermedio(x, y)
        vector_cargar(y + (x - 1) * num_columnas) = cargar(x, y)
        vector_posicion(y + (x - 1) * num_columnas) = posicion(x, y)
    Next y
Next x

a = 1
For b = 1 To multiplicador
    For x = 1 To R
        For y = 1 To num_columnas
            If a < contador * num_columnas Then
                nuevo_intermedio(x, y + (b - 1) * num_columnas) = vector_intermedio(a)
                nuevo_cargar(x, y + (b - 1) * num_columnas) = vector_cargar(a)
                nuevo_posicion(x, y + (b - 1) * num_columnas) = vector_posicion(a)
                a = a + 1
            End If
        Next y
    Next x
Next b

contador = R

Else
multiplicador = 1
For x = 1 To contador
    For y = 1 To num_columnas
        nuevo_intermedio(x, y) = intermedio(x, y)
        nuevo_cargar(x, y) = cargar(x, y)
        nuevo_posicion(x, y) = posicion(x, y)
    Next y
Next x
End If
```



```
For x = 1 To contador
  For y = 1 To multiplicador * num_columnas
    mejor_matriz(x, y) = nuevo_intermedio(x, y)
    mejor_cargar(x, y) = nuevo_cargar(x, y)
    mejor_posicion(x, y) = nuevo_posicion(x, y)
  Next y
Next x

For x = 1 To contador
For a = 1 To multiplicador
  For y = (a - 1) * num_columnas + 1 To a * (num_columnas) - 1
    min_tiempo(x, a) = min_tiempo(x, a) + tdist(nuevo_intermedio(x, y) + 1,
nuevo_intermedio(x, y + 1) + 1)
  Next y
Next a
Next x

For x = 1 To contador
For a = 1 To multiplicador
  min_tiempo(x, a) = min_tiempo(x, a) + tdist(1, nuevo_intermedio(x, (a - 1) * num_columnas
+ 1) + 1) + tdist(nuevo_intermedio(x, a * num_columnas) + 1, 1)
Next a
Next x

For c = 1 To multiplicador
For d = 1 To contador
  For f = 1 To 1
    For b = 1 To c * num_columnas
      guardar_aleat(b) = 0
      nuevo(b) = 0
    Next b

    a = (c - 1) * num_columnas + 1
    For x = 1 To 6 * M
      If a <= c * num_columnas Then
        aleatorio = Int(((c * num_columnas) - ((c - 1) * num_columnas + 1) + 1) * Rnd + (c - 1) *
num_columnas + 1)
        If guardar_aleat(aleatorio) <> aleatorio Then
          nuevo(a) = aleatorio
          a = a + 1
          guardar_aleat(aleatorio) = aleatorio
        End If
      End If
    Next x

    suma_tiempo(d, c) = 0

    For x = (c - 1) * num_columnas + 1 To c * num_columnas - 1
      suma_tiempo(d, c) = suma_tiempo(d, c) + tdist(nuevo_intermedio(d, nuevo(x)) + 1,
nuevo_intermedio(d, nuevo(x + 1)) + 1)
    Next x
```

```

suma_tiempo(d, c) = suma_tiempo(d, c) + tdist(1, nuevo_intermedio(d, nuevo((c - 1) *
num_columnas + 1)) + 1) + tdist(nuevo_intermedio(d, nuevo(c * num_columnas)) + 1, 1)

```

```

If suma_tiempo(d, c) < min_tiempo(d, c) And suma_tiempo(d, c) <> 0 Then

```

```

    min_tiempo(d, c) = suma_tiempo(d, c)
    For x = (c - 1) * num_columnas + 1 To c * num_columnas
        mejor_matriz(d, x) = nuevo_intermedio(d, nuevo(x))
        mejor_cargar(d, x) = nuevo_cargar(d, nuevo(x))
        mejor_posicion(d, x) = nuevo_posicion(d, nuevo(x))
    Next x
End If
Next f
Next d
Next c

```

'Inventariar'

```

ReDim llegada(3 * M, multiplicador * J * (L - 1))
ReDim llegada_acum(3 * M, multiplicador * J * (L - 1))
ReDim tiempo_mio(contador, multiplicador * num_columnas)

```

```

For x = 1 To tiempo
    For y = 1 To multiplicador * J * (L - 1)
        llegada(x, y) = 0
        llegada_acum(x, y) = 0
    Next y
Next x

```

```

For x = 1 To contador
    For y = 1 To multiplicador * num_columnas
        tiempo_mio(x, y) = 0
    Next y
Next x

```

```

For y = 1 To J * (L - 1)
    llegada(1, y) = stock_inicial(y)
Next y

```

```

For x = 1 To contador
    tiempo_mio(x, 1) = tdist(1, mejor_matriz(x, 1) + 1)
Next x

```

```

For x = 1 To contador
    For y = 1 To (multiplicador * num_columnas) - 1
        tiempo_mio(x, y + 1) = tiempo_mio(x, y) + tiempo_mio(x, y) + tdist(mejor_matriz(x, y)
+ 1, mejor_matriz(x, y + 1) + 1)
    Next y
Next x

```

```

For x = 1 To contador
  For a = 1 To multiplicador
    For y = (a - 1) * num_columnas + 1 To a * (num_columnas)
      If y <= multiplicador * num_columnas Then
        llegada(tiempo_mio(x, y), mejor_posicion(x, y)) = llegada(tiempo_mio(x, y),
mejor_posicion(x, y)) + mejor_cargar(x, y)
      End If
    Next y
  Next a
Next x

For y = 1 To J * (L - 1)
  llegada_acum(1, y) = llegada(1, y)
Next y

For x = 2 To tiempo
  For y = 1 To J * (L - 1)
    llegada_acum(x, y) = llegada_acum(x - 1, y) + llegada(x, y)
  Next y
Next x

demanda_sin_servir = 0
For y = 1 To J * (L - 1)
  If llegada_acum(1, y) - transporte(1, y) < 0 Then
    demanda_sin_servir = demanda_sin_servir + (transporte(1, y) - llegada_acum(1, y))
    llegada_acum(1, y) = 0
  Else
    llegada_acum(1, y) = llegada_acum(1, y) - transporte(1, y)
  End If
Next y

For x = 2 To tiempo
  For y = 1 To J * (L - 1)
    If llegada_acum(x - 1, y) + llegada(x, y) - transporte(x, y) < 0 Then
      demanda_sin_servir = demanda_sin_servir + (transporte(x, y) - (llegada_acum(x - 1, y) +
llegada(x, y)))
      llegada_acum(x, y) = 0
    Else
      llegada_acum(x, y) = llegada_acum(x - 1, y) + llegada(x, y) - transporte(x, y)
    End If
  Next y
Next x

'Calcular costes'
coste_dist = 0
For x = 1 To contador
  For y = 1 To multiplicador * num_columnas - 1
    coste_dist = coste_dist + (TC * tdist(mejor_matriz(x, y) + 1, mejor_matriz(x, y + 1) + 1))
  Next y
Next x

```

```
For x = 1 To contador
    coste_dist = coste_dist + tdist(1, mejor_matriz(x, y) + 1)
Next x

total_cargar = 0
For x = 1 To contador
    For y = 1 To num_columnas
        total_cargar = total_cargar + (mejor_matriz(x, y)) * MC
    Next y
Next x

coste_inv = 0
For x = 1 To tiempo
    For y = 1 To (L - 1)
        For a = 1 To J
            coste_inv = coste_inv + (llegada_acum(x, -J + y * J + a) * HC(a))
        Next a
    Next y
Next x

'Que se quede con la mejor solución'
coste_total = coste_dist + total_cargar + (contador * coste_vehiculo) + coste_inv +
demanda_sin_servir * 1000

If coste_total < sol_coste_total Then

    sol_coste_total = coste_total

    For x = 1 To L - 1
        sol_ruta(x) = ruta(x)
    Next x

    sol_multiplicador = multiplicador

    sol_coste_dist = coste_dist
    sol_total_cargar = total_cargar
    sol_contador = contador
    sol_coste_inv = coste_inv
    sol_demanda_sin_servir = demanda_sin_servir
    sol_num_columnas = num_columnas

ReDim sol_mejor_matriz(sol_contador, sol_multiplicador * sol_num_columnas)

    For x = 1 To contador
        For y = 1 To multiplicador * sol_num_columnas
            sol_mejor_matriz(x, y) = mejor_matriz(x, y)
        Next y
    Next x

End If
Next w
```

'Mostrar en pantalla la mejor solución'

```
Worksheets("Inicio").Cells(11, 4).Value = sol_coste_dist
Worksheets("Inicio").Cells(11, 5).Value = sol_total_cargar
Worksheets("Inicio").Cells(11, 6).Value = sol_contador * coste_vehiculo
Worksheets("Inicio").Cells(11, 7).Value = sol_coste_inv
Worksheets("Inicio").Cells(11, 8).Value = sol_demanda_sin_servir * 1000
Worksheets("Inicio").Cells(11, 9).Value = sol_coste_total
Worksheets("Inicio").Cells(11, 11).Value = sol_contador
```

```
For x = 1 To sol_contador
```

```
  a = 1
```

```
  Worksheets("Rutas").Cells(3 + x, 4).Value = 0
```

```
  If sol_num_columnas > 1 Then
```

```
    For b = 1 To sol_multiplicador
```

```
      For y = (b - 1) * sol_num_columnas + 1 To b * sol_num_columnas - 1
```

```
        If (sol_mejor_matriz(x, y) <> 0 Or (sol_mejor_matriz(x, y) = 0 And sol_mejor_matriz(x, y + 1) <> 0)) And sol_mejor_matriz(x, y) <> sol_mejor_matriz(x, y + 1) Then
```

```
          Worksheets("Rutas").Cells(3 + x, a + 4).Value = sol_mejor_matriz(x, y)
```

```
          a = a + 1
```

```
        End If
```

```
      Next y
```

```
      If sol_mejor_matriz(x, b * sol_num_columnas) = 0 And Worksheets("Rutas").Cells(3 + x, a + 3).Value <> 0 Then
```

```
        Worksheets("Rutas").Cells(3 + x, a + 4).Value = 0
```

```
        a = a + 1
```

```
      End If
```

```
      If sol_mejor_matriz(x, b * sol_num_columnas) <> 0 And Worksheets("Rutas").Cells(3 + x, a + 3).Value <> 0 Then 'esto lo pongo ya que antes solo m imprimirá hasta sol_mejor_matriz-1
```

```
        Worksheets("Rutas").Cells(3 + x, a + 4).Value = sol_mejor_matriz(x, y)
```

```
        Worksheets("Rutas").Cells(3 + x, a + 5).Value = 0
```

```
        a = a + 2
```

```
      End If
```

```
    Next b
```

```
  Else
```

```
    Worksheets("Rutas").Cells(3 + x, 5).Value = sol_mejor_matriz(x, 1)
```

```
    Worksheets("Rutas").Cells(3 + x, 6).Value = 0
```

```
  End If
```

```
Next x
```

```
Worksheets("Rutas").Cells(2, 5).Value = "Localizaciones (L=0=warehouse)"
```

```
For x = 1 To sol_contador
```

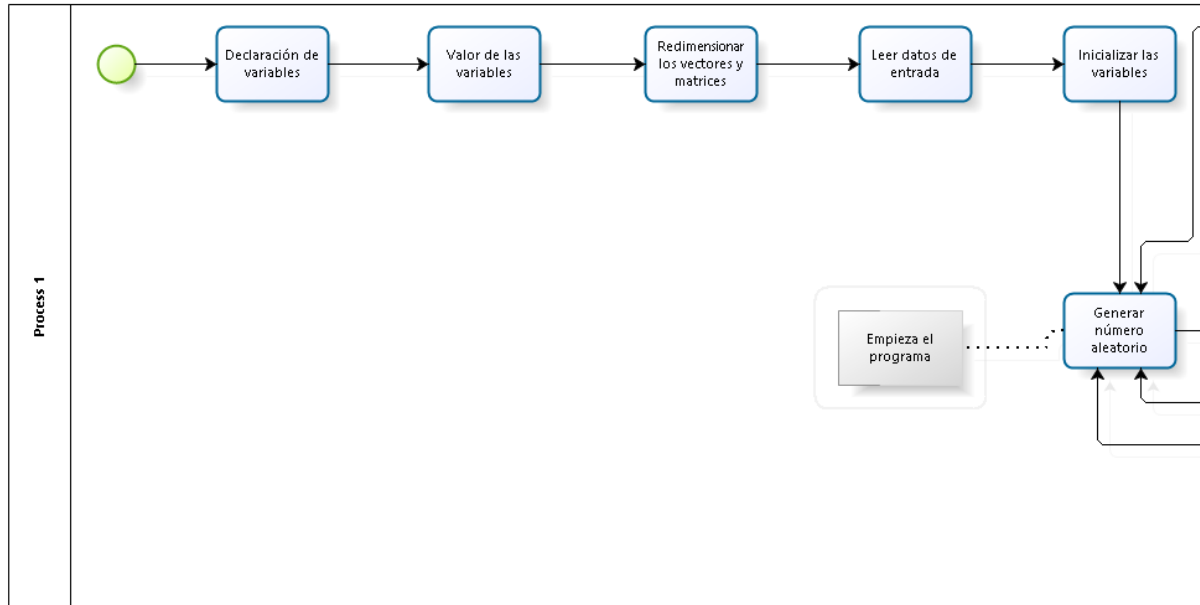
```
  Worksheets("Rutas").Cells(x + 3, 2).Value = "Carretilla"
```

```
  Worksheets("Rutas").Cells(x + 3, 3).Value = x
```

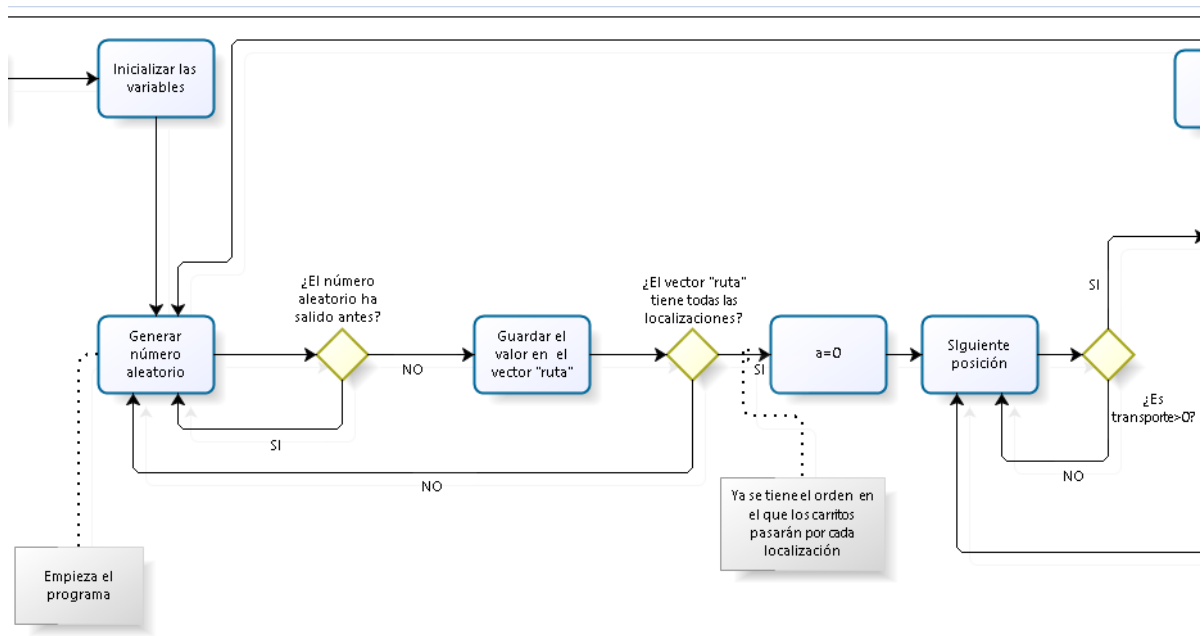
```
Next x
```

```
End Sub
```

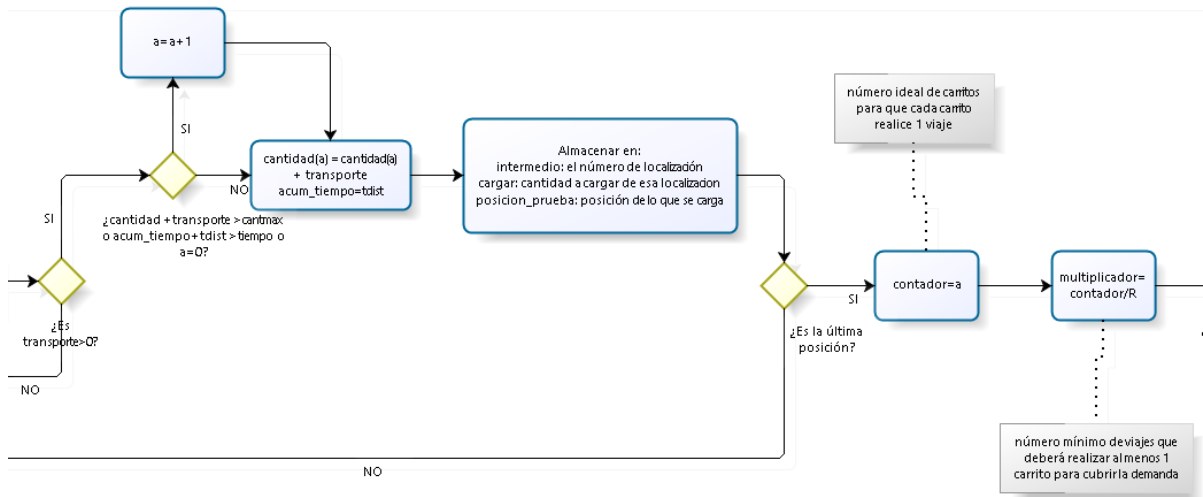
8.3 Diagramas de Flujo del Programa



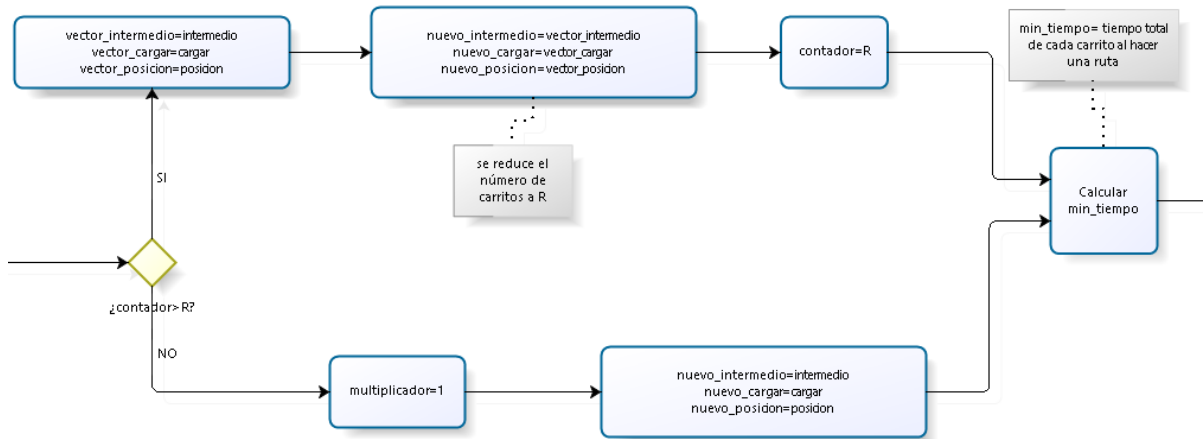
Anexo 1 – Declaración, valor y dimensión de las variables



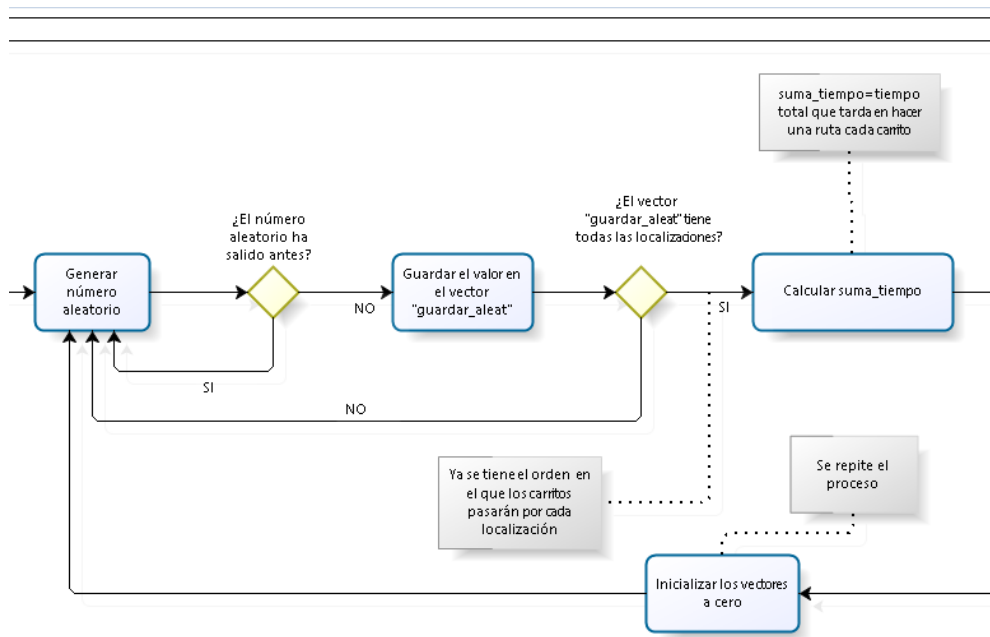
Anexo 2 – Creación de una ruta provisional aleatoria



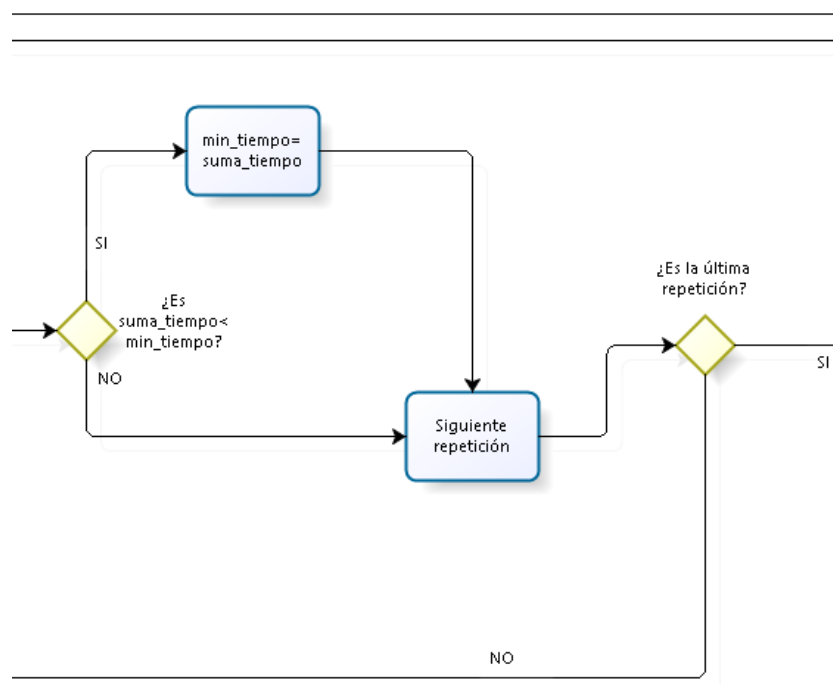
Anexo 3 – Cantidad a transportar en cada viaje



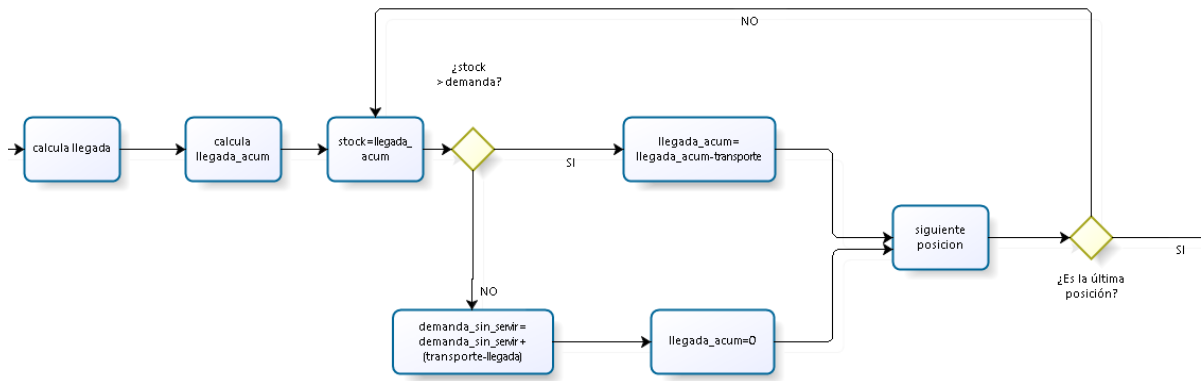
Anexo 4 – Que solo hayan R número de rutas



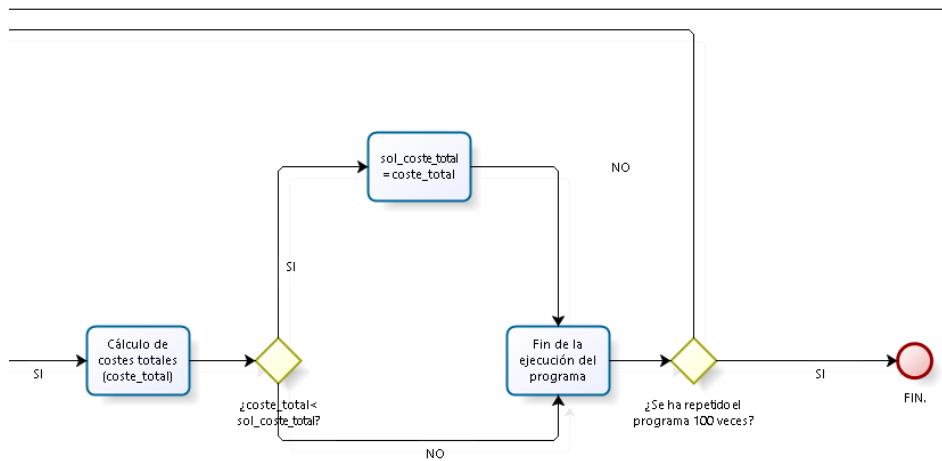
Anexo 5 - Que solo hayan R número de rutas



Anexo 6 - Que solo hayan R número de rutas



Anexo 7 - Inventariar



Anexo 8 – Calculo de los costes totales

8.4 Índices de Imágenes, Diagramas, Tablas y Gráficas.

Nota: Se omite la referencia a la fuente cuando se trata de una elaboración propia.

Índice de imágenes

Imagen 1. Imagen de 1910 donde se observa que el vehículo está estático y son los operarios los que se mueven para realizarle las operaciones necesarias. (Imagen extraída de Henry Ford, Mi vida y obra, 1925.)

Imagen 2. Línea de ensamblaje en Ford. Los operarios siempre están en la misma posición y son los vehículos los que están en movimiento mediante una cinta transportadora. (Imagen extraída de Alvolante info, 2016)

Imagen 3. Vista de toda la planta de Ford Almussafes. (Imagen extraída de autobild.es, 2016)

Imagen 4. A la izquierda se observa la manera en la que se suministraba al cliente anteriormente, mientras que a la derecha se muestra el suministro mediante el Milk-run. (Elaboración propia a partir de Nippon Express, 2016)

Imagen 5. Entregas individuales.

Imagen 6. Entregas por Milk-run.

Imagen 7. Carretilla recogepedidos. (Imagen extraída de Jungheinrich)

Imagen 8. Carretilla elevadora retráctil. (Imagen extraída de Interempresas Media, S.L.)

Imagen 9. Transpaleta manual. (Imagen extraída de Yale)

Imagen 10. Situación inicial en la planta en la que el camión descarga el material y la carretilla elevadora lo carga.

Imagen 11. Una vez la carretilla elevadora ha cargado el material, ésta se dispone a dejarlo en la línea correspondiente.

Imagen 12. Una vez la carretilla elevadora ha cargado el material, ésta se dispone a dejarlo en la línea correspondiente.

Imagen 13. Resolución mediante CoinMP.

Imagen 14. Resolución mediante LPSolve.

Imagen 15. Carretilla 1. Ruta 1. (L=3, J=2, R=2)

Imagen 16. Carretilla 1. Ruta 1. (L=3, J=2, R=2)

Imagen 17. Como crear una macro desde Microsoft Office Excel 2010.

Imagen 18. Como crear una macro desde cualquier versión anterior a Microsoft Office Excel

Imagen 19. Ventanas de la macro.

Imagen 20. Pantalla principal de Excel.

Imagen 21. Diseño exterior.

Imagen 22. Experimento 1. Inicio.

Imagen 23. Experimento 1. Rutas.

Imagen 24. Experimento 1. Carretillero 1. Ruta 1.

Imagen 25. Experimento 1. Carretillero 1. Ruta 2.

Imagen 26. Experimento 2. Carretillero 2. Ruta 1.

Imagen 27. Experimento 2. Carretillero 2. Ruta 2.

Imagen 28. Experimento 2. Inicio.

Imagen 29. Experimento 2. Rutas.

Imagen 30. Experimento 3. Inicio.

Imagen 31. Experimento 3. Rutas.

Imagen 32. Experimento 4. Inicio.

Imagen 33. Experimento 4. Rutas.

Imagen 34. Experimento 5. Inicio.

Imagen 35. Experimento 5. Rutas.

Imagen 36. Experimento 6. Inicio.

Imagen 37. Experimento 6. Rutas.

Imagen 38. Experimento 7. Inicio.

Imagen 39. Experimento 7. Rutas.

Imagen 40. Experimento 8. Inicio.

Imagen 41. Experimento 8. Rutas.

Imagen 42. Experimento 9. Inicio.

Imagen 43. Experimento 9. Rutas.

Imagen 44. Experimento 10. Inicio.

Imagen 45. Experimento 10. Rutas.

Imagen 46. Experimento 11. Inicio.

Imagen 47. Experimento 11. Rutas.

Imagen 48. Experimento 12. Inicio.

Imagen 49. Experimento 12. Rutas.

Imagen 50. Experimento 13. Inicio.

Imagen 51. Experimento 13. Rutas.

Imagen 52. Experimento 14. Inicio.

Imagen 53. Experimento 14. Rutas.

Índice de diagramas

Diagrama 1. Representación de las diferentes secciones que del área de Fabricación.

Diagrama 2. Representación de los diferentes modelos que se producen en cada línea.

Diagrama 3. Representación de una cadena de suministro.

Diagrama 4. El ciclo de Berry, Davis (1996) representando los pasos a seguir al realizar un modelo matemático.

Diagrama 5. Ciclo en cascada. Winston Royce en 1970 y revisada por Barry Boehm.

Índice de tablas

Tabla 1. Producciones aproximadas/modelo y día. (Información proporcionada por Ford.)

Tabla 2. Definición de los parámetros del modelo.

Tabla 3. Tiempos de desplazamiento.

Tabla 4. Matriz de demanda.

Tabla 5. Stock inicial.

Tabla 6. Coste de almacenaje.

Tabla 7. Ejemplo con variable entera.

Tabla 8. Ejemplo con variable binaria.

Tabla 9: Resolución de las rutas ($L=3$, $J=2$, $R=2$).

Tabla 10: Tipo de variables y sus rangos.

Tabla 11: Nomenclatura. Valor de las variables.

Tabla 12: Nomenclatura. Ejecutar el programa 100 veces.

Tabla 13: Nomenclatura. Crear una ruta provisional aleatoria.

Tabla 14: Nomenclatura. Cantidad a transportar en cada viaje.

Tabla 15: Nomenclatura. Multiplicador.

Tabla 16: Nomenclatura. Que sólo haya como máximo R número de rutas (I).

Tabla 17: Nomenclatura. Que sólo haya como máximo R número de rutas (II).

Tabla 18. Nomenclatura. Inventariar.

Tabla 19. Nomenclatura. Calcular costes

Tabla 20. Nomenclatura. Que se quede con la mejor solución posible.

Índice de gráficos

Gráfico 1. Evolución de las ventas de automóviles en España. (Extraído de Suzuki).



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

PRESUPUESTO

DESARROLLO Y ANÁLISIS DE UNA HERRAMIENTA DE CÁLCULO DE RUTAS DE TRANSPORTE A CAPACIDAD FINITA PARA ABASTECER A UNA LÍNEA DE MONTAJE DE AUTOMÓVILES

AUTORA: LORENA BAVIERA FORNER

TUTOR: CARLOS ANDRÉS ROMANO

Curso académico: 2015-16

ÍNDICE PRESUPUESTO

9. PRESUPUESTO	
9.1. Presupuesto de Inversión.....	104
9.2. Presupuesto de Explotación.....	105
9.3. Rentabilidad Económica.....	107
9.4. Conclusiones.....	110
10. ANEXOS DEL PRESUPUESTO.....	111

9. PRESUPUESTO

El presupuesto del presente proyecto consta de dos partes:

1. Presupuesto de inversión
2. Presupuesto de explotación

9.1. Presupuesto de Inversión

El presupuesto de inversión es la cantidad de dinero necesaria para la obtención del objeto del proyecto. Esta inversión que hay que realizar se puede ver reflejada en la siguiente gráfica (curva de la S):

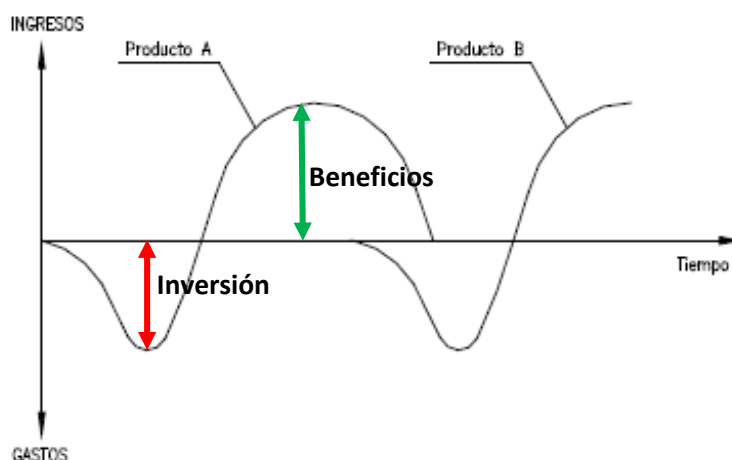


Gráfico 2. Curva de la S.

En la gráfica anterior, se puede observar que al principio, la realización de un proyecto en vez de suponer un beneficio, supone un gasto. Pero, la realidad es que para poder obtener algún resultado positivo, primeramente hay que realizar una inversión, ya sea esta inversión puramente económica o por esfuerzo y horas dedicadas en la realización del mismo.

Si el proyecto está bien definido y es realmente útil, en un plazo de tiempo se obtendrán los beneficios y la recuperación de la inversión inicial, como bien se puede observar en la gráfica de arriba. El plazo de tiempo en el que se logrará dependerá de la repercusión que tenga el proyecto en la empresa y de la magnitud de la inversión.

En el presupuesto de inversión se incluyen los siguientes gastos:

Visita a la empresa Ford Almussafes:

Para poder conocer bien la planta y el funcionamiento de la misma, se decidió realizar una visita a Ford. Esto supone un coste dado que se ha invertido un tiempo tanto en ir, como en la realización de la visita.

Número de visitas realizadas	Coste/viaje (€)	Coste total (€)
1	4,00	4,00

Tabla 21. Costes visita a Ford Almussafes.

Análisis y formación en las herramientas usadas

Aunque antes de realizar el TFG, el alumno ya había aprendido brevemente a realizar modelos matemáticos durante la carrera, nunca aprendió a utilizar una herramienta para poder programar en ella, con lo cual, se desconocía por completo el funcionamiento de MPL. Para poder aprender a utilizar MPL, se realizaron unos tutoriales con un profesional en la materia de la Universidad Politécnica de Valencia.

Número de reuniones	Horas/reunión (h/reunión)	Número total de horas (h)	Coste/hora (€/h)	Coste total (€)
2	3	6	30,00	180,00

Tabla 22. Costes aprendizaje MPL.

Además de reunirse con un profesional del MPL, el alumno también se reunió en múltiples ocasiones con su profesor tutor para que éste le explicase o corrigiese ciertos puntos del proyecto.

También fue el mismo tutor quien proporcionó al alumno un mayor conocimiento sobre la herramienta Visual Basic, que ya había conocido durante la carrera, pero no con tanta profundidad.

	Número de reuniones	Horas/reunión (h/reunión)	Número total de horas (h)	Coste/hora (€/h)	Coste total (€)
Reunión Visual Basic	3	3	9	30,00	270,00
Reunión TFG	9	1	9	30,00	270,00

Tabla 23. Costes aprendizaje VBA.

Con esto, se termina el presupuesto de inversión.

Este tendría un coste de: $4,00 + 180,00 + 270,00 + 270,00 = 724,00€$

9.2. Presupuesto de explotación

El presupuesto de explotación es una previsión sobre la explotación del objeto del proyecto. Aquí se incluirán costes tales como costes de mano de obra, materiales y otros.

Realización de las herramientas:

Este paso se puede subdividir en tres:

- i. Construcción del modelo

	Número total de horas (h)	Coste/hora (€/h)	Coste total (€)
Modelo matemático (MPL)	60	4,50	270,00
Algoritmo (Visual Basic)	130	4,50	585,00

Tabla 24. Costes construcción del modelo.

ii. Experimentación

	Número total de horas (h)	Coste/hora (€/h)	Coste total (€)
Modelo matemático (MPL)	10	4,50	45,00
Algoritmo (Visual Basic)	20	4,50	90,00

Tabla 25. Costes experimentación.

iii. Evaluación de los resultados

	Número total de horas (h)	Coste/hora (€/h)	Coste total (€)
Modelo matemático (MPL)	15	4,50	67,50
Algoritmo (Visual Basic)	30	4,50	135,00

Tabla 26. Costes de evaluación de los resultados.

Se ha considerado el coste por hora en estas actividades realizadas por el alumno de 4,50€. Un alumno trabajando como estudiante en prácticas cobraría alrededor de 800,00€/mes, lo que equivale a 4,50€/hora.

Materiales:

Para la realización del TFG se han utilizado varios materiales además de un portátil para poder desarrollar tanto las herramientas (VBA y MPL) como la redacción del TFG.

En cuanto a los materiales necesarios, éstos son los siguientes:

- Impresora
- Tinta para la impresora
- Fotocopias
- Encuadernación del TFG
- Bolígrafos
- Libreta

	Unidades	Coste/unidad (€)	Coste total (€)
Portátil	1	700,00	700,00
Material de oficina (varios)	6	-	120,00

Tabla 27. Costes materiales.

Redacción del TFG:

Por último, una vez se han desarrollado todas las herramientas y se cuenta con la información suficiente, el alumno puede proceder a la redacción del trabajo. Este proceso es complejo y también incluye la realización de una presentación que posteriormente tendrá que exponer ante un tribunal.

	Horas dedicadas (h)	Coste/hora (€/h)	Coste total (€)
Redacción TFG	120	4,50	540,00

Tabla 28. Costes redacción TFG.

Teniendo en cuenta todos los costes del presupuesto de explotación, éste tendría un coste total de: $270,00 + 585,00 + 45,00 + 90,00 + 67,50 + 135,00 + 700,00 + 120,00 + 540,00 = 2.552,50€$.

Coste total del proyecto:

Una vez ya se conocen ambos presupuestos, se deduce que el presupuesto total será la suma de ambos. A continuación se pueden ver en una tabla todos los costes considerados, tanto de inversión como de explotación y el coste total que supondría este trabajo.

	Costes (€)
Presupuesto de inversión:	-
Visita a la empresa	4,00
Análisis y formación en herramientas usadas	720,00
Presupuesto de explotación:	-
Realización de las herramientas	1.192,50
Materiales	820,00
Redacción del TFG	540,00
Total:	3.276,50

Tabla 29. Coste total del proyecto.

El coste total del proyecto es de: 3.276,50 € (tres mil doscientos setenta y seis euros con cincuenta.)

9.3. Rentabilidad del Proyecto

Para saber si un proyecto es conveniente llevarlo a cabo, primero debe ser evaluado y para ello, se calcula la rentabilidad del mismo. Si el proyecto saliera rentable, éste se pondría en marcha en cuanto se pudiese, mientras que si no lo fuera, habría que rechazar el proyecto.

A la hora de calcular la rentabilidad del proyecto se utilizarán tres indicadores:

- Pay-back (plazo de recuperación)
- Valor Actualizado Neto (VAN)
- Tasa de Rentabilidad Interna (TIR)

Se ha decidido emplear estos tres indicadores pues son los más utilizados a la hora de decidir si un proyecto es o no es viable. Hay que tener en cuenta que mediante estos indicadores se predecirá el éxito o fracaso del proyecto, pero aun saliendo el proyecto rentable, esto no garantiza que en el futuro el proyecto vaya a serlo, ya que todo se basa en estimaciones.

Pay-back:

El Payback o Plazo de Recuperación es un método estático para la evaluación de inversiones y se define como el periodo de tiempo requerido para recuperar el capital inicial de una inversión.

$$PB = \frac{\text{Inversión inicial}}{\text{Beneficio promedio anual}}$$

Como beneficio promedio anual se utilizará el beneficio mínimo que se espera de este proyecto. Al encontrar mejores rutas por donde el carretillero deberá suministrar a cada línea, la cantidad de stock será menor, y esto hará que los costes de inventariar disminuyan. Se considera que esto supondrá como mínimo unos 30.000€/año.

Para calcular la amortización hay que dividir los costes de las instalaciones por el número de periodos a considerar. En este caso, amortización = (720+100) / 5 = 164,00 € / año.

Mediante esta tabla se muestran los cálculos y pasos realizados para la resolución del Pay-Back.

	Año 0	Año 1	Año 2	Año 3	Año 4	Año 5
Amortización (€)		164,00	164,00	164,00	164,00	164,00
Beneficio bruto (€)		30.000,00	30.000,00	30.000,00	30.000,00	30.000,00
Impuestos 35% (€)		10.500,00	10.500,00	10.500,00	10.500,00	10.500,00
Beneficio Neto (€)		19.500,00	19.500,00	19.500,00	19.500,00	19.500,00
Inversión Inicial (€)	-3276,50					
FC (€)		19.336,00	19.336,00	19.336,00	19.336,00	19.336,00

Tabla 30. Cálculo del pay-back.

$$PB = \frac{\text{Inversión inicial}}{\text{Beneficio promedio anual}} = \frac{3276.50}{19500} = 0,17 \text{ años}$$

Un Pay-Back de 0,17 años representa el tiempo que se tardará en recuperar la inversión realizada. Esto significa que aproximadamente a los dos meses de poner en marcha este proyecto, se recuperaría la inversión, lo cual es positivo ya que es un periodo de tiempo muy corto.

A pesar de que el “índice de Payback” es fácil de calcular e interpretar, tiene el inconveniente de no tener en consideración cualquier beneficio o pérdida que se pueda obtener posteriormente al periodo de recuperación. Por ello, se procede a calcular el VAN.

VAN (Valor Actual Neto):

El Valor Actual Neto es una medida de la rentabilidad absoluta neta que proporciona el proyecto. Se basa en el cálculo del valor actual de un número futuro de flujos de caja, que surgen mediante una inversión. Si el VAN>0, entonces el proyecto se considerará como rentable, en cambio, siempre que sea menor a cero, no interesará realizar este proyecto. Que el VAN sea mayor a cero simplemente implica que el proyecto, teniendo en cuenta todos los gastos y la inversión, producirá más beneficios que gastos, con lo que será rentable la realización del mismo.

$$VAN = -I_0 + \sum \frac{FNC_t}{(1+k)^t}$$

Donde:

- I_0 : Inversión inicial
- FNC_t : Flujos Netos de Caja en cada periodo de tiempo
- k : Tasa de interés

Para calcular la rentabilidad de este proyecto se consideran los siguientes datos:

- $I_0 = 3276,50 \text{ €}$
- $k = 3,00 \%$
- $t = 5 \text{ años}$

$$VAN = -I_0 + \sum \frac{FNC_t}{(1+k)^t} = -3276.5 + \frac{19336}{(1+0.03)^5} = 13.402,90 \text{ €}$$

El resultado para el VAN es de: 13.402,90€

Como VAN=13.402,90€>0, el proyecto es rentable.

TIR:

La Tasa Interna de Retorno o de Rentabilidad (TIR), es un método de valoración de inversiones que mide la rentabilidad de los cobros y los pagos actualizados, generados por una inversión, en términos relativos.

$$VAN = -I_0 + \sum \frac{FNC_t}{(1+k)^t} = 0 \rightarrow VAN = 0$$

Para sacar la TIR, el VAN se iguala a cero. Mediante este método, se conseguirá averiguar el valor de k . El proyecto será rentable siempre que $k >$ tipo de interés actual (3%). La TIR siempre debe ser mayor al tipo de interés actual, pues esto significa que se obtiene un mayor beneficio con este proyecto que si se decidiese invertirlo en el banco con el tipo de interés actualmente existente.

Despejando k de la fórmula, se obtiene que TIR=0.4262 = 42,62%

Como TIR=42,62%>tipo interés actual (3%), resultaría beneficioso invertir en este proyecto.

9.4. Conclusiones:

Como la inversión que se realiza para llevar a cabo el proyecto es mínima comparada con otros costes que tiene Ford, en realidad, no haría falta calcular la rentabilidad de este trabajo mediante los métodos básicos como son el Pay-back, VAN y la TIR, pues se sabe que obteniendo tan solo un beneficio anual mayor a 3.276,50€ al utilizar las herramientas diseñadas, este proyecto sería rentable. De cualquier modo, el cálculo efectuado por los tres métodos de viabilidad de la inversión resultan satisfactorios; por tanto, la inversión en este proyecto es rentable.

Es lógico pensar que esto ocurrirá ya que de momento se sigue un procedimiento de suministrar a las líneas cuando se ve que no hay stock, sin ningún tipo de análisis, mientras que ahora sí que lo habrá. Es evidente que habrá ocasiones en las que lo que haría el carretillero y lo que dice la herramienta informática que se debe hacer, coinciden, pero en muchos otros momentos, será menos costoso seguir las rutas que manda el programa. Mientras esta situación se repita en varias ocasiones, los costes disminuirán, con lo que la inversión se verá cubierta y el proyecto empezará a ser rentable.

Añadiendo a lo anterior, no sólo se debe tener en consideración la rentabilidad del proyecto, sino que hay que tener también en cuenta el riesgo que conlleva la implementación del mismo. En ocasiones, un proyecto puede tener un gran riesgo de aceptación, si por ejemplo se tratara de cambiar el diseño de un automóvil, éste podría finalmente no agradar a los clientes. Ese tipo de proyectos con altos riesgos hay que valorarlos con más detenimiento. Pero, en este caso, el riesgo de que este proyecto fracase es prácticamente nulo, y por ello, habría que seguir contando con su realización.

10. ANEXOS DEL PRESUPUESTO

Nota: Se omite la referencia a la fuente cuando se ha trata de una elaboración propia

Índice de tablas

Tabla 21. Costes visita a Ford Almussafes.

Tabla 22. Costes aprendizaje MPL.

Tabla 23. Costes aprendizaje VBA.

Tabla 24. Costes construcción del modelo.

Tabla 25. Costes experimentación.

Tabla 26. Costes de evaluación de los resultados.

Tabla 27. Costes materiales.

Tabla 28. Costes redacción del TFG.

Tabla 29. Coste total del proyecto.

Tabla 30. Cálculos del Pay-Back.

Índice de gráficos

Gráfico 2. Curva de la S. (Elaboración propia a partir de Gestión de proyectos, Proyectos, GIOI (UPV)).