



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería en Organización Industrial**

# **Determinación del número de tarjetas en circulación en la metodología Kanban**

**Autor:**

**Doral Yagüe, Fernando**

**Tutor(es):**

**Araúzo Araúzo, José Alberto  
Departamento de Organización de  
Empresas y CIM**

**Valladolid, Julio 2023**



## RESUMEN

El sistema Kanban es una metodología que comenzó a implantarse a mediados del siglo pasado en Japón. Su gran objetivo es reducir el trabajo en curso y optimizar cada uno de los procesos de una fábrica. La metodología Kanban persigue que la producción se realice en función a una demanda real en cada instante: sistema pull. Todo ello nos lleva, además, a una reducción en los inventarios, ya que solo se produce lo que se necesita en ese instante. En este contexto, en este trabajo de fin de grado (TFG) se estudia un resumen de todo tipo de herramientas (metaheurísticas, métodos puramente matemáticos, etc.) que existen hoy en día para obtener el número óptimo de tarjetas Kanban. Además, se ha desarrollado en Microsoft Excel una simulación de una línea de producción para poder comprobar de primera mano cómo puede influir el número de tarjetas y otros parámetros.

## PALABRAS CLAVE

Tarjeta Kanban, trabajo en curso, contenedores, centro de trabajo, Just In Time.

## **ABSTRACT**

Kanban System is a methodology that began to be implemented in the middle of the last century in Japan. Its great goal is to reduce work in process and optimize each of the processes of a factory. Kanban methodology pursues that production is carried out based on current demand (pull system). It's a great change, because formerly production was carried out based on future demand (push system). It causes a reduction in the inventory level, because it is only produced what is demanded in that moment. In this context, in this Final Degree Project (TFG), a summary of all the methods used to calculate the optimal number of Kanban cards and the optimal lot size is studied. In addition, I have developed a simulation of a production line that implements Kanban system using Microsoft Excel to discuss all the benefits of Kanban System.

## **KEY WORDS**

Kanban card, Work in Process (WIP), containers, work centre, Just in Time.

# Índice

RESUMEN.....	1
PALABRAS CLAVE.....	1
ABSTRACT.....	2
KEY WORDS .....	2
ÍNDICE DE ILUSTRACIONES.....	5
ÍNDICE DE TABLAS.....	7
1. INTRODUCCIÓN .....	9
1.1 CONTEXTO DEL ESTUDIO.....	9
1.2 JUSTIFICACIÓN .....	9
1.3 OBJETIVOS.....	10
1.4 CONTENIDO DE LA MEMORIA .....	10
2. LEAN MANUFACTURING: SISTEMA KANBAN .....	11
2.1 LEAN MANUFACTURING: ANTECEDENTES .....	11
2.2 TOYOTA PRODUCTION SYSTEM.....	11
2.3 DEFINICIÓN.....	12
2.4 SISTEMAS PUSH Y PULL .....	13
2.5 JUST IN TIME: SISTEMA KANBAN .....	14
2.6 TARJETAS KANBAN .....	15
2.6.1 TIPOS DE TARJETAS KANBAN .....	19
3. HERRAMIENTAS CÁLCULO DE TARJETAS.....	21
3.1 MÉTODOS METAHEURÍSTICOS.....	21
3.1.1 RECOCIDO SIMULADO .....	22
3.1.2 ALGORITMO MEMÉTICO .....	28
3.1.3 ALGORITMO GENÉTICO MULTI OBJETIVO.....	38
3.2 MÉTODOS MATEMÁTICOS .....	47
3.2.1 DIMENSIONAMIENTO DEL BÚFER PARA UN SISTEMA HEIJUNKA KANBAN.....	47
3.2.2 PROGRAMACIÓN DINÁMICA PARA UNA PRODUCCIÓN DE LÍNEA DE UN PRODUCTO CON VARIAS ETAPAS CONTROLADA POR KANBAN .....	57
3.3 MÉTODOS DE SIMULACIÓN .....	63
3.3.1 ESTUDIO DE SIMULACIÓN EPQ Y KANBAN .....	63
4. MODELO DE SIMULACIÓN.....	69

4.1	INTRODUCCIÓN Y ELECCIÓN DE SIMULADOR.....	69
4.2	MODELO PRELIMINAR .....	70
4.2.1	RESULTADOS Y CONCLUSIONES .....	82
4.3	MEJORA DEL MODELO.....	89
4.3.1	RESULTADOS Y CONCLUSIONES .....	93
5.	CONCLUSIONES Y TRABAJOS FUTUROS .....	103
5.1	TRABAJOS FUTUROS .....	104
6.	BIBLIOGRAFÍA .....	105

## ÍNDICE DE ILUSTRACIONES

Ilustración 1: Historia de la Manufactura. Fuente: (Socconini, 2019) .....	11
Ilustración 2: Representación Sistema push. Fuente: (Sendil Kumar & Paneerselvam, 2007).....	14
Ilustración 3: Funcionamiento del área de los Kanban de Retirada. Fuente: (Esparrajo Jr, 1988).....	17
Ilustración 4: Funcionamiento del área de los Kanban de producción. Fuente: (Esparrajo Jr, 1988).....	18
Ilustración 5: Representación de un sistema Kanban completo. Fuente: (Esparrajo Jr, 1988).....	19
Ilustración 6: Ilustración de cómo el recocido simulado permite el salto también desde óptimos. (Peredo, 2006).....	22
Ilustración 7: Pasos a seguir en un algoritmo memético. Fuente: (Priyom Dutta, 2022).....	28
Ilustración 8: Esquema de funcionamiento de la línea de producción. Fuente: (Rabbani, Layegh, & Ebrahim, 2009).....	31
Ilustración 9: Estructura principal del algoritmo memético. Fuente: (Rabbani, Layegh, & Ebrahim, 2009).....	34
Ilustración 10: Esquema del funcionamiento de la recombinación. Fuente: (Rabbani, Layegh, & Ebrahim, 2009).....	35
Ilustración 11: Value Stream mapping para la producción del techo solar. Fuente: (Hou & Hu, 2011) .....	42
Ilustración 12: Modelo de simulación de la planta. Fuente: (Hou & Hu, 2011) .....	43
Ilustración 13: Funcionamiento del tablero heijunka. Fuente: (Matzka, Di Mascolo, & Furmans, 2012) .....	49
Ilustración 14: Funcionamiento milk run. Fuente: (Brar & Saini, 2011) .....	50
Ilustración 15: Modelo propuesto de un sistema heijunka Kanban. Fuente: (Matzka, Di Mascolo, & Furmans, 2012) .....	51
Ilustración 16: Resumen de resultados. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012).....	60
Ilustración 17: Modelo utilizado. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012) ...	62
Ilustración 18: Representación EPQ. Fuente: (Moran & Brayer, 2011) .....	65
Ilustración 19: Resultados Annual setup cost vs setup time. Fuente: (Moran & Brayer, 2011).....	67
Ilustración 20. Modelo de simulación.....	70
Ilustración 21: Botones del modelo de simulación.....	71
Ilustración 22: Código de botón de inicio. ....	72
Ilustración 23: Código botón de parada. ....	72
Ilustración 24: Código botón de reinicio. ....	73
Ilustración 25: Código macro Kanban .....	73
Ilustración 26: Representación del CT1.....	75

Ilustración 27: Representación del CT2.....	79
Ilustración 28: Representación CT3.....	80
Ilustración 29: % Productividad de los 3 centros de trabajo en función del número de tarjetas Kanban en CT3. ....	84
Ilustración 30: % Productividad en los 3 centros de trabajo en función del número de tarjetas Kanban en CT1. ....	85
Ilustración 31: % Productividad en los 3 centros de trabajo en función del número de tarjetas Kanban en CT1. ....	86
Ilustración 32: Trabajo en curso en función del número de tarjetas Kanban. ....	87
Ilustración 33: % Productividad en CT3 en función del tiempo entre envíos.....	88
Ilustración 34: Representación modelo mejorado.....	90
Ilustración 35: Representación CT3 y CT4. ....	93
Ilustración 36: Representación posible distribución en una línea con 6 centros de trabajo. ....	94
Ilustración 37: Representación del modelo aislando a CT1.....	95
Ilustración 38: Representación del modelo aislando a CT3.....	96
Ilustración 39: Representación del modelo con todos los trabajos en funcionamiento.....	97
Ilustración 40: % de Productividad en los distintos centros de trabajo en función del número de Kanban en CT1.....	98
Ilustración 41: % de Productividad en función del número de tarjetas Kanban en CT4.....	99
Ilustración 42: Evolución del trabajo en curso en función del número de tarjetas Kanban.....	100

## ÍNDICE DE TABLAS

Tabla 1: Parámetros elegidos para el algoritmo. Fuente: (Shahabudeen, Gopinath, & Krishnaiah, 2002) .....	26
Tabla 2: Resultados del algoritmo. Fuente: (Shahabudeen, Gopinath, & Krishnaiah, 2002).....	27
Tabla 3: Parámetros del sistema utilizados. Fuente: (Rabbani, Layegh, & Ebrahim, 2009).....	36
Tabla 4: Tabla de resultados. Fuente: (Rabbani, Layegh, & Ebrahim, 2009) .....	37
Tabla 5: Soluciones de los óptimos de Pareto. Fuente: (Hou & Hu, 2011).....	46
Tabla 6: Resultados comparativos con SIMAN. Fuente: (Matzka, Di Mascolo, & Furmans, 2012).....	54
Tabla 7: Resultados cambiando número de Kanban. Fuente: (Matzka, Di Mascolo, & Furmans, 2012).....	55
Tabla 8: Número de Kanban para cada nivel de servicio. Fuente: (Matzka, Di Mascolo, & Furmans, 2012).....	56
Tabla 9: Efecto en el aumento de número de servidores para la etapa número 3. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012) .....	61
Tabla 10: Efecto del tamaño del contenedor en los resultados de la etapa 3. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012) .....	61
Tabla 11: Resultados para el cambio de número de servidores, en esta ocasión para el modelo de simulación. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012).....	62
Tabla 12: Variables utilizadas en la simulación. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012) .....	65
Tabla 13: Parámetros aplicables para EPQ y Kanban. Fuente: (Moran & Brayer, 2011).....	66
Tabla 14: Variables utilizadas en EPQ. Fuente: (Moran & Brayer, 2011).....	66
Tabla 15: Variables utilizadas en Kanban. Fuente: (Moran & Brayer, 2011).....	66
Tabla 16: Celdas donde se especifica la velocidad, y se muestra el reloj del proceso por pantalla.....	71
Tabla 17. Celdas G2 y G3. ....	72
Tabla 18: Celdas donde se establece el número de tarjetas y el tiempo de ciclo. ..	74
Tabla 19: Celdas donde establecemos el tiempo de ciclo y nos muestra el reloj de los envíos. ....	75
Tabla 20: Resultados primera simulación.....	83
Tabla 21: Parámetros utilizados en la simulación del modelo preliminar. ....	88
Tabla 22: Ejemplo de matriz de flujo. ....	89
Tabla 23: Celdas correspondientes a cada elemento de la matriz de flujo. ....	90
Tabla 24: Matriz de flujo en la hoja de Excel.....	92
Tabla 25: Representación de la matriz de flujo aislando a CT1. ....	94
Tabla 26: Configuración de la matriz de flujo aislando a CT3.....	96
Tabla 27: Configuración final de la matriz de flujo. ....	97
Tabla 28: Valores utilizados en la experimentación en el modelo mejorado. ....	98

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

# 1. INTRODUCCIÓN

## 1.1 CONTEXTO DEL ESTUDIO

La gestión eficiente de inventarios y la optimización de todos los procesos de producción es clave para poder garantizar la competitividad y el éxito de cualquier empresa, y más en el contexto empresarial en el que nos encontramos, con la irrupción de la Industria 4.0.

La metodología Kanban, que fue desarrollada por primera vez por Toyota en sus fábricas de Japón, ha demostrado su efectividad a lo largo de los años, aumentando la productividad en todo tipo de sectores, a pesar de que se implementara en un primer momento en fábricas de automóviles.

El sistema Kanban para control de inventarios se basa en el principio de control visual, utilizándose para gestionar el flujo de trabajo y reducir lo máximo posible los niveles de inventario en una empresa. Como veremos a continuación, se basa en la filosofía de los cinco ceros: cero stocks, cero defectos, cero averías, cero plazos y cero papeles, consiguiéndose así una mejora continua de todos los procesos de una línea de producción.

Mediante el uso de las tarjetas Kanban como unidades de control, el objetivo principal es equilibrar la producción con la demanda real, es decir, tener un sistema pull que produce en función de la demanda real, y no un sistema push, que produce en función de las demandas futuras. Todo esto nos permite evitar la sobreproducción en la fábrica, lo que dispararía nuestros costes de inventario, y por supuesto evita que se produzcan cuellos de botella, o al menos disminuirlos lo máximo posible.

## 1.2 JUSTIFICACIÓN

La gestión de inventarios es un quebradero de cabeza para todas las empresas, y en este contexto uno de los desafíos que se muestran en este Trabajo de Fin de Grado (TFG) es la determinación de la cantidad óptima de tarjetas en circulación, para con ello tener un flujo de trabajo lo más eficiente posible.

El número de tarjetas debe estar equilibrado, es decir, tenemos que evitar que sea muy bajo, debido a que conlleva a interrupciones en la producción y a escasez de materiales. También se debe procurar que no sea un número excesivo, debido a que se puede acumular trabajo en curso en grandes cantidades. Esto lo que provocaría sería una acumulación de productos y una congestión a nivel productivo.

La determinación del número óptimo de tarjetas en circulación es crucial, para que sea acorde con la filosofía Just In Time, reduciendo los tiempos de espera entre procesos y maximizando así la capacidad de respuesta ante cambios en la demanda. Esta tarea puede llegar a ser muy complicada ya que hay una gran variabilidad dentro de los procesos productivos que no podemos evitar, además de la incertidumbre en la demanda.

Por ello, en este Trabajo de Fin de Grado (TFG) se propone abordar esta problemática, desarrollando un modelo de simulación que permita conseguir el número óptimo de tarjetas Kanban en una línea de producción, para así tener un buen flujo de materiales e intentar reducir los costes de inventarios.

### 1.3 OBJETIVOS

El gran objetivo de este TFG es, primero, tener una recopilación de diversas técnicas para determinar el número de tarjetas Kanban en circulación y el tamaño de lote óptimo exponiendo su metodología y analizando sus resultados.

Por otro lado, y como se ha expuesto en el anterior apartado, se ha realizado un modelo de simulación con la herramienta Microsoft Excel y las macros correspondientes. Con ello se ha querido hacer una pequeña representación visual del flujo de tarjetas a lo largo de un proceso productivo en línea con el Trabajo de Fin de Grado.

### 1.4 CONTENIDO DE LA MEMORIA

La memoria va a constar aparte del presente capítulo de introducción, de los siguientes apartados:

- Lean Manufacturing: Sistema Kanban: El capítulo va a tratar del fundamento teórico del sistema Kanban como una breve introducción a los apartados siguientes
- Herramientas de cálculo de tarjetas: En este apartado se ha hecho una recopilación en la literatura acerca de distintas técnicas para la obtención del número de tarjetas Kanban
- Modelo de simulación: Trata sobre la realización de una simulación de un sistema Kanban con el objetivo de estudiar su comportamiento y la influencia del número de tarjetas en circulación

Una vez concluidos los capítulos ya mencionados este trabajo se concluirá con unas conclusiones y una colección de las todas las referencias bibliográficas utilizadas en el documento.

## 2. LEAN MANUFACTURING: SISTEMA KANBAN

### 2.1 LEAN MANUFACTURING: ANTECEDENTES

Antiguamente, las cadenas de producción eran totalmente rudimentarias y apenas estaban automatizadas. Hasta finales del siglo XVIII con el desarrollo de la máquina de vapor, comenzarían a modernizarse paulatinamente todas las fases de producción de cualquier producto. En la línea del tiempo realizada por (Socconini, 2019) se puede apreciar mejor:

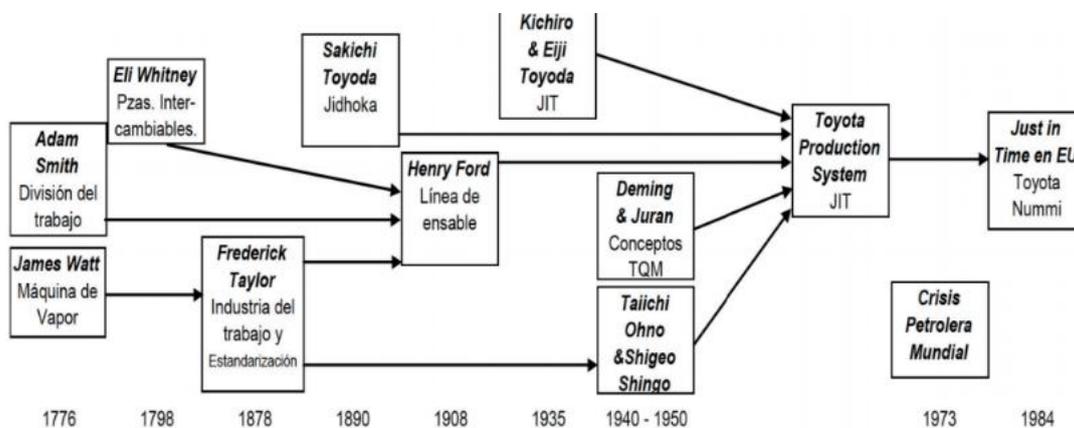


Ilustración 1: Historia de la Manufactura. Fuente: (Socconini, 2019)

Entre las figuras más importantes antes de la irrupción del modelo de Toyota están Frederick Taylor y Henry Ford.

Frederick Taylor introdujo el sistema de producción por lotes a finales del siglo XIX, siendo un modelo para la industria occidental, proponiendo la estandarización del trabajo.

Por su parte, Henry Ford aplicó en la fase de producción de los primeros vehículos de Ford los principios de Adam Smith. Estos principios se basaban en la división del trabajo en tareas, lo que su introducción en la producción del modelo T de Ford revolucionó la forma de trabajar en estas plantas.

### 2.2 TOYOTA PRODUCTION SYSTEM

El sistema de producción de Toyota ("Toyota Production System" en inglés), a día de hoy conocido como Just in Time, tiene su origen en Japón. Al finalizar la Segunda Guerra Mundial, el país estaba devastado, y tenían la necesidad de volver a poner en funcionamiento su industria.

Japón hizo grandes esfuerzos en recuperar la competitividad mundial a nivel industrial, traduciéndose en el empeño de los grandes empresarios japoneses en modernizar y hacer mucho más competitivas sus empresas.

El presidente de Toyota en la época, Kiichiro Toyoda, decidió crear un sistema que le aportara gran rentabilidad, ya que vio que los empleados japoneses eran mucho menos competitivos que los occidentales.

Más tarde, cuando estaba al frente de la compañía Eiji Toyoda y de la mano de Taiichi Ohno, tuvieron un gran éxito internacional gracias a la implantación del Just In Time. Esto supuso un gran aumento de la rentabilidad y la asociación con la empresa americana General Motors.

## 2.3 DEFINICIÓN

El término Lean Manufacturing es el término que recibe el Just In Time en occidente. Su gran objetivo es el desarrollo de un proceso continuo de identificación y eliminación de desperdicios o excesos, ya que esto no genera valor, y lo único que consigue es crear sobrecostes.

Esto supone una continua búsqueda de procesos de mejora dentro de una empresa, ya que nunca se va a conseguir ser productivo al 100% y siempre habrá aspectos que se puedan mejorar: mejora continua (Kaizen).

Kaizen proviene de dos palabras japonesas que son: “Kai”, que significa cambio, y “Zen”, que significa algo parecido a mejora. Según (Tapias & Correa, 2010) se basa en dos pilares básicos, como son los equipos de trabajo y la ingeniería industrial, empleados para mejorar los sistemas productivos.

Su objetivo es la estandarización de procesos y el incremento de la productividad, controlando todos los procesos dentro de una planta de producción. Esto se consigue con la reducción de los tiempos de ciclo, la estandarización de los criterios de calidad, y la eliminación de desperdicios, denominado muda.

Ofrece una gran cantidad de beneficios, siendo los siguientes un breve resumen de todos ellos:

- Reducción de inventarios
- Disminución de la cantidad de accidentes
- Reducción de los tiempos de preparación de todas las máquinas (setup time)
- Aumento de los niveles de satisfacción de clientes y consumidores
- Incrementos de productividad
- Reducción de costes
- Menores niveles de desperdicios y despilfarro
- Capacidad de competir en mercados globalizados

Para la implementación de la mejora continua, (Socconini, 2019) apunta que hay tres grandes limitantes de la productividad, que son conocidos como los 3 “Mu:

- Muri: Sobrecarga
- Mura: Variabilidad
- Muda: Desperdicio

**Sobrecarga o Muri:** Si queremos mejorar la productividad, uno de los aspectos que hay que tener en cuenta es la sobrecarga, ya que, si se sobrepasa cierta capacidad, va a suponer una caída en el rendimiento de trabajadores y máquinas.

**Variabilidad o Mura:** La estandarización de la producción se produce con una buena uniformidad de los procesos. En el caso de no haber uniformidad y encontrarnos con variabilidad en los productos finales, es importante determinar las causas. Cuando son causas naturales se dice que el proceso está bajo control y que no supone ningún problema. En cambio, si se produce una gran variación, el proceso se considera fuera de control.

**Desperdicio o Muda:** Uno de los grandes objetivos del Lean Manufacturing es la eliminación de estos desperdicios, que como se ha mencionado justo antes, reducen la productividad. Un desperdicio o exceso es cualquier esfuerzo realizado por una empresa que no sea del todo esencial para agregar valor al producto. Todo ello reduce el nivel de servicio, habiendo varios tipos:

- Muda de sobreproducción
- Muda de sobreinventario
- Muda de productos defectuosos
- Muda de transporte de materiales y herramientas
- Muda de procesos innecesarios
- Muda de espera
- Muda de movimientos innecesarios del trabajador

Dentro de todos ellos, cabe destacar la sobreproducción, considerado el peor desperdicio. Esto es debido a que el mantenimiento de los inventarios es muy costoso, reduciendo los recursos de la empresa. Lo ideal sería no tener inventarios, pero a medida que se acumula más inventario, evidencia los problemas de una fase de producción, ya que hay producción estancada.

## 2.4 SISTEMAS PUSH Y PULL

Ambos sistemas son dos enfoques distintos a la hora de gestionar una cadena de suministro dentro de una empresa

En el sistema push, como se va a ver con la ilustración de (Sendil Kumar & Panneerselvam, 2007), la producción y la distribución se planifican en función de la demanda futura, mandándose los productos a los distribuidores sin tener en cuenta la demanda actual. Cuando en una estación de trabajo se termina cierta actividad, se manda a la siguiente estación de trabajo donde necesita almacenamiento o continuar con el procesamiento. A medida que pasa el tiempo, la tarjeta de trabajo

pasa de una etapa a otra, pero esto supone un gran problema cuando surgen cambios impredecibles, como por ejemplo cuando se produce una avería o un retraso, se acumula inventario de trabajo en curso.



*Ilustración 2: Representación Sistema push. Fuente: (Sendil Kumar & Paneerselvam, 2007)*

Por otro lado, en el sistema pull la producción se planifica en función de la demanda real en ese momento que solicita el mercado, entregándose los productos en función de las órdenes recibidas. Este sistema es mucho más flexible que el anterior, ya que puede adaptarse al mercado de una manera mucho más ágil y rápida. En este caso hay varias estaciones de trabajo, y cuando se termina un trabajo es retirado por la estación de trabajo siguiente. Esto supone un gran cambio respecto al sistema anterior, ya que el trabajo es “tirado” por la estación siguiente, en vez de ser “empujado” a la estación de trabajo consecutiva.

El flujo de partes a través de la línea de producción es controlado por tarjetas Kanban. Estas tarjetas pueden ser de tarjeta única o de tarjeta doble y cada estación de trabajo tiene un stock de entrada y un stock de salida.

La principal ventaja que proporciona este sistema respecto al sistema push es que reduce el coste de inventario y por lo tanto reduce también el coste asociado al inventario.

## 2.5 JUST IN TIME: SISTEMA KANBAN

El sistema JIT (Just In Time ) fue desarrollado por el ya mencionado Taiichi Ohno, en el llamado Toyota Production System. En la década de los 50, las compañías japonesas comenzaron a operar con bajos niveles de inventario y teniendo grandes niveles de calidad y productividad. Esto es debido a que el sistema Just In Time sigue la filosofía de los cinco ceros: cero defectos, cero colas, cero inventarios, cero averías, etc.

Con ello se asegura la llegada del número correcto de piezas en el lugar y tiempo adecuados. Además, el sistema JIT se convierte en algo esencial en una planta de producción. Esto se debe a que el avance de la tecnología hizo que el sistema tradicional de producción push quedara obsoleto. Los planes de producción de materiales (MRP) habían sido sustituidos por el sistema Just In Time (método pull), donde el trabajo en curso puede ser supervisado de una manera mucho más precisa que en los métodos anteriores.

El sistema Kanban juega un papel fundamental dentro de los sistemas de producción JIT. Kanban es una tarjeta en la que viene toda la información necesaria para la

producción o ensamblaje de un producto concreto. Este sistema se puede implementar en plantas de más de una etapa (“multistage” en inglés), para la planificación de la producción y control del inventario. Con el sistema Kanban se facilita según (Sendil Kumar & Panneerselvam, 2007) un mayor volumen de producción y la reducción de los tiempos de preparación y de trabajo en curso. Pero antes, de todo, se va a resaltar la diferencia entre los sistemas de tipo push y los sistemas de tipo pull.

## COMPONENTES KANBAN

Este sistema fue desarrollado por Toyota, para cumplir los objetivos del Just In Time. Para conocer su funcionamiento, es necesario determinar cuáles son sus componentes.

- Tarjetas Kanban
- Contenedores
- Centros de trabajo
- Buzones Kanban de producción y transporte
- Zonas de entrada
- Zonas de salida

## 2.6 TARJETAS KANBAN

En el sistema de producción de Toyota se utilizaban dos tipos de tarjetas: tarjetas de producción y de transporte. La función de esa tarjeta de transporte es autorizar el movimiento de un contenedor que tiene ya un número de piezas o partes de un centro de trabajo al siguiente.

Según (Esparrajo Jr, 1988) una tarjeta de transporte al uso consta de cinco especificaciones:

- Descripción de la parte o pieza: Contiene información de la pieza que está siendo transportada, como su número, dimensiones y tamaño, etc.
- Tamaño del contenedor: Especifica el número exacto de piezas que hay en su interior
- Número Kanban de retirada: Indica a qué contenedor se ha asociado la tarjeta Kanban, y también cuantos contenedores se han utilizado en ese centro de trabajo.
- Centro de trabajo previo: Describe el procedimiento anterior y otra información relevante acerca del proceso.
- Próximo centro de trabajo: Este último elemento proporciona información acerca de la próxima etapa.

Para el otro tipo de tarjeta, la tarjeta de producción, el autor sostiene que debe llevar seis especificaciones:

- Descripción de la pieza: Información necesaria de la pieza para continuar con su fabricación, al igual que en el Kanban de transporte.

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

- Tamaño del contenedor: Nos indica cuantas partes serán producidas en ese determinado contenedor.
- Descripción del centro de trabajo: Información acerca del centro de trabajo y descripción del proceso que producirá esa parte o pieza.
- Localización de almacenamiento: Indica dónde el contenedor será almacenado una vez que se hayan las partes requeridas se hayan producido.
- Requerimientos de material: Componentes necesarios para producir dicho material.

Tanto las tarjetas de producción como las de transporte pueden contener mucha más información como códigos de barras, tiempos de llegada y de salida, o firmas de autorización del proceso.

### COMPONENTES DEL ÁREA DE PRODUCCIÓN

En un sistema Kanban, el área de producción es el lugar donde se producen todos los ítems. El punto central es el centro de trabajo, donde se llevan a cabo todo tipo de tareas, ya sea de manera manual, con máquinas o con robots totalmente automatizados.

Buzón de Kanban de producción: Es el lugar donde se autoriza la producción de elementos en una línea de producción

### CONTENEDORES KANBAN

Los contenedores utilizados en nuestro sistema Kanban pueden ser cualquiera que pueda almacenar un cierto número autorizado de partes. Es conveniente que estén estandarizados para que todos los centros de trabajo puedan hacer uso de los mismos contenedores.

### ZONAS DE ENTRADA Y SALIDA

La zona de entrada se refiere al área dedicada al almacenamiento de stock que va a ser utilizado por el centro de trabajo y a todo el material requerido para poder producir la próxima parte. El contenedor y su tarjeta asociada no se mueven hasta que todos los componentes hayan sido movidos al centro de trabajo.

La zona de salida se refiere como el propio nombre indica, al área dedicada al almacenamiento de productos terminados que están preparados para ser enviados al cliente. Estos productos terminados se almacenan en un contenedor para productos terminados, y el contenedor no se envía hasta que no haya sido llenado con un número de productos terminados determinado.

### COMPONENTES DEL ÁREA DE RETIRADA

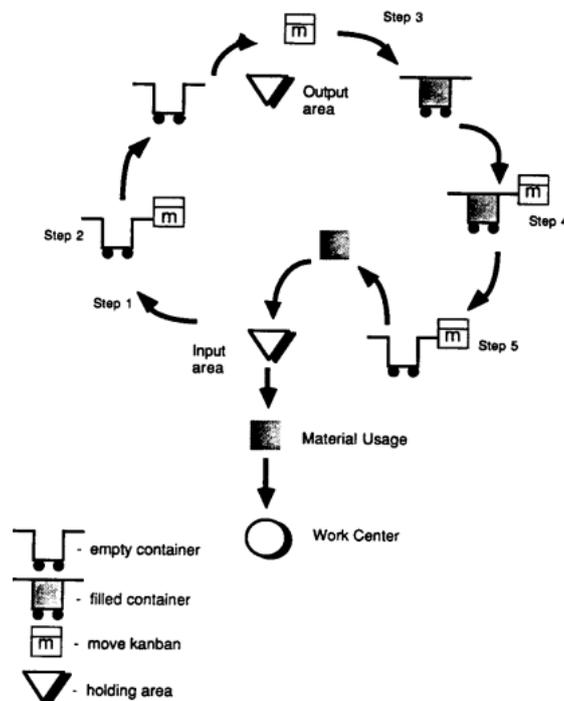
El buzón Kanban de transporte sirve para almacenar las tarjetas de transporte. Una vez que se utilizan todas las piezas de un contenedor, el centro de trabajo manda el contenedor vacío y mueve la tarjeta con su Kanban de retirada correspondiente. Esta tarjeta indica al buzón Kanban que el centro de trabajo tiene que ser repuesto, y el buzón autoriza al Kanban de transporte a ir asociado al contenedor vacío. El

contenedor vacío entonces, es enviado al área de salida del centro de trabajo anterior.

### ÁREA DE KANBANS DE RETIRADA

Según (Esparrajo Jr, 1988), el ciclo operativo del Kanban de retirada actúa de la siguiente manera, apreciándose a continuación gráficamente:

1. Un centro trabajo cualquiera utiliza los materiales provistos para realizar una operación o tarea prevista. Desde que el contenedor está vacío, da una señal al buzón Kanban de transporte para solicitar y asociar un Kanban de transporte a dicho contenedor
2. Después de asociar la tarjeta al contenedor, el operario transporta el contenedor al área de salida del centro de trabajo anterior.
3. En el área de salida el operario retira la tarjeta del contenedor vacío a un contenedor que esté lleno, que ha tenido que ser llenado por el Kanban de producción del centro de trabajo anterior. El operario retira también la tarjeta de producción del contenedor lleno y la deja en el contenedor vacío.
4. El operario mueve el contenedor lleno desde el área de salida al área de entrada del próximo centro de trabajo
5. El centro de trabajo ha sido repuesto con nuevos materiales necesarios para completar el producto. Una vez el centro de trabajo usa todo el material del contenedor, el ciclo comienza de nuevo.



*Ilustración 3: Funcionamiento del área de los Kanban de Retirada. Fuente: (Esparrajo Jr, 1988)*



Entre los beneficios más importantes se encuentra la reducción del trabajo en curso, reduciendo su inventario a un nivel donde un centro de trabajo esté trabajando con un contenedor solamente. Todo esto se consigue reduciendo o el número o el tamaño de los contenedores. A continuación, tenemos una representación de (Esparrajo Jr, 1988) sobre un esquema de su funcionamiento:

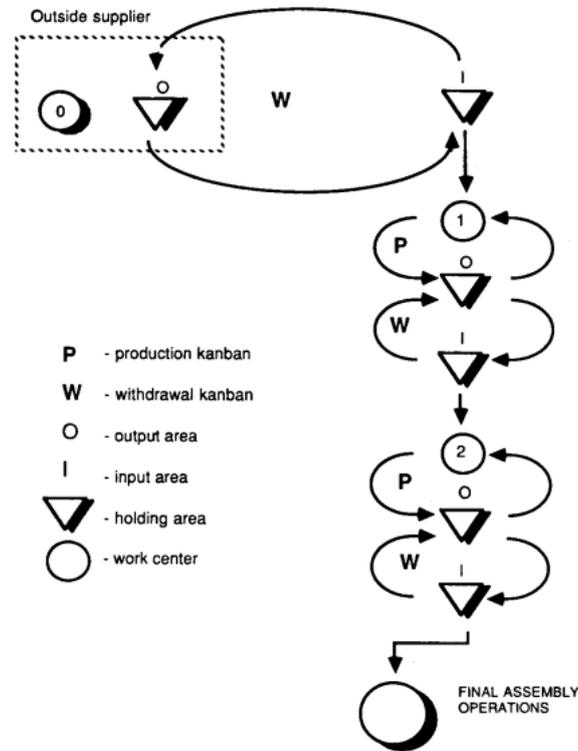


Ilustración 5: Representación de un sistema Kanban completo. Fuente: (Esparrajo Jr, 1988)

Otra de las ventajas que tiene Kanban es que involucra a los trabajadores en su participación, mejorando su moral y su productividad. Las decisiones sobre cuantos contenedores y de qué tamaño tienen que ser pueden ser tomadas por los propios trabajadores de la fábrica.

Por otro lado, una revisión constante del sistema permite a la compañía tener mejoras continuas en su proceso.

### 2.6.1 TIPOS DE TARJETAS KANBAN

Sistema de doble tarjeta: Es el tipo más conocido y el más utilizado. Aparte de Toyota, otras compañías lo utilizan, como Nissan Motor, que utiliza un sistema parecido denominado Action Plate Method (APM).

Sistema de tarjeta única: En este sistema solo se utiliza el Kanban de retirada o de transporte. Es una combinación entre un sistema push y pull. Las piezas se realizan de acuerdo con una planificación, pero las reposiciones de material a los centros de trabajo son autorizadas por las tarjetas de transporte. Las partes continúan hacia la

producción con el sistema push, mientras que los centros de trabajo obtienen sus provisiones con un sistema pull.

### MRP sincronizado

Yamaha creó un sistema que combina el sistema Kanban con los MRP, dando lugar al MRP sincronizado (“Syncho MRP” en inglés). Se desarrolló para un gran volumen de producción que constara de varios productos diferentes.

La compañía prepara una planificación para el ensamblaje final, mientras que el MRP es utilizado para generar la planificación de la producción en los distintos centros de trabajo. Estos calendarios de producción autorizan dos tipos de tarjetas:

- Synchro I: Parecida al Kanban de transporte
- Synchro II: Similar al Kanban de producción

Estas tarjetas están hechas específicamente para cada centro de trabajo, y son una salida de un ordenador que también genera una planificación diaria para cada centro de trabajo.

### Micro Kanban

Para finalizar, tenemos otro método que combina MRP y Just In Time. En este caso se añade un módulo de fabricación (Kanban) en el ya instalado sistema MRP. MRP se utilizará para planificar la compra y distribución de materiales, y luego se utilizará un microordenador para mostrar por pantalla el ritmo y el número de partes disponibles en la línea de ensamblado.

Esa misma información se imprime y se distribuye como tarjetas Kanban de producción y de transporte. Los datos procesados se envían desde un ordenador central a microcomputadores cerca de los centros de trabajo, que mandarían a su vez información al ordenador central.

El microordenador entonces autorizará Kanbans de producción o de transporte, según vayan siendo necesarios, siendo un buzón Kanban electrónico.

A continuación, el ordenador central mandará de vuelta la información al sistema MRP. Entonces, el sistema MRP utilizará esa información para volver a planear o planificar las operaciones.

### CONCLUSIONES

A raíz del artículo de (Esparrajo Jr, 1988) se ha podido ir explicando cómo funciona el sistema Kanban desde sus inicios a mediados del siglo pasado, y su funcionamiento y puesta en marcha. Este sistema revolucionó la industria japonesa y más tarde fue adoptado por países occidentales como Estados Unidos, lo que demuestra su eficiencia y productividad.

## 3.HERRAMIENTAS CÁLCULO DE TARJETAS

En este apartado se va a desarrollar un estado del arte acerca de todos los tipos y metodologías que se han ido desarrollando para el cálculo del número de tarjetas Kanban desde los años 50 hasta la actualidad.

El "estado del arte" es un término utilizado para describir el nivel más avanzado de desarrollo en un campo específico de conocimiento o tecnología. Este término se utiliza comúnmente en la investigación académica para describir la revisión exhaustiva de la literatura existente en un área temática, que incluye los desarrollos más recientes y los avances más significativos.

Hay infinidad de autores y se ha decidido resumir todas estas técnicas en función de los métodos que han utilizado a partir de la clasificación de (Guerrero Collazos, 2014).

- Metaheurística
- Matemático
- Simulación

### 3.1 MÉTODOS METAHEURÍSTICOS

Se entiende como heurística al conjunto de técnicas para resolver problemas, en nuestro caso la obtención del número óptimo de tarjetas Kanban.

El estudio de las técnicas que se van a ver a continuación corresponde a métodos metaheurísticos, es decir, un método que parte de una heurística para resolver un problema determinado utilizando distintos parámetros en el modelo, siguiendo unos procedimientos que comúnmente son heurísticos.

Entre las técnicas más conocidas de aplicación de métodos metaheurísticos se encuentran la búsqueda tabú, el enjambre de abejas, etc.

En este apartado se va a hacer referencia a tres artículos científicos distintos, en los que se aplican un método metaheurístico distinto para el objetivo principal, que es la obtención del número óptimo de tarjetas Kanban en circulación.

Los artículos que se han mencionado utilizan las técnicas metaheurísticas que se van a presentar:

- Recocido simulado
- Algoritmo memético
- Algoritmo genético

A continuación, se va a ir haciendo una explicación de cada uno de estos métodos, fijándonos en el procedimiento utilizado y los resultados obtenidos.

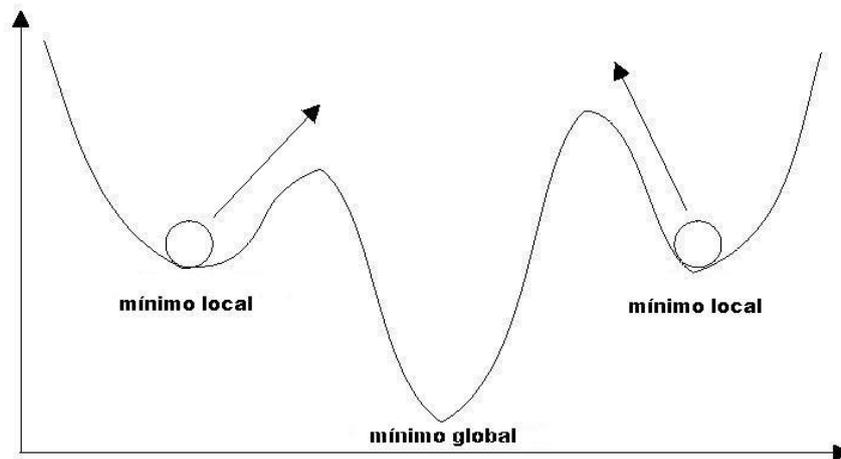
### 3.1.1 RECOCIDO SIMULADO

El recocido simulado (simulated annealing en inglés) es un algoritmo de optimización heurística utilizado para encontrar soluciones a problemas de optimización global.

Este algoritmo se basa en el proceso de recocido en metalurgia, donde un material se calienta y luego se enfría lentamente para obtener una estructura cristalina óptima.

En el recocido simulado, se genera una solución al azar y se evalúa su calidad mediante una función objetivo. Luego, se realizan iteraciones en las que se aceptan nuevas soluciones incluso si empeoran la función objetivo, lo que permite escapar de óptimos locales. La probabilidad de aceptar una solución peor se reduce gradualmente con el tiempo para evitar que el algoritmo se atasque en soluciones de baja calidad.

El recocido simulado puede ser utilizado en una amplia variedad de problemas de optimización, como el problema del viajante, el problema de la mochila, el problema de asignación cuadrática, entre otros. Aunque no garantiza encontrar la solución óptima, el recocido simulado puede proporcionar soluciones de alta calidad en tiempos razonables para problemas de gran tamaño. Ahora se va a poder ver una imagen esquemática acerca de su funcionamiento proporcionada por (Peredo, 2006).



*Ilustración 6: Ilustración de cómo el recocido simulado permite el salto también desde óptimos. (Peredo, 2006)*

Este algoritmo se puede utilizar en infinidad de áreas como en la optimización de las rutas de transporte, con el famoso problema del viajante, en la planificación de recursos, etc.

## MÉTODO

En el modelo que se va a explicar a continuación realizado por (Shahabudeen, Gopinath, & Krishnaiah, 2002), el objetivo principal es determinar el número de tarjetas Kanban en cada estación y el tamaño de lote óptimo utilizando la técnica del recocido simulado.

Se trata de un modelo de simulación de tarjeta simple y una función objetivo con dos criterios distintos. Esto último es algo inusual ya que en la mayoría de las herramientas se hace solo con un criterio.

### PRIMER CRITERIO

El primer criterio se basa en la maximización de la tasa de rendimiento media acumulada, que está definida por el cociente entre la demanda cumplida y la demanda generada.

### SEGUNDO CRITERIO

El segundo criterio está centrado en la minimización de la suma de la cola Kanban promedio, que es la suma del número medio de Kanban esperando en cada estación.

#### Tasa de rendimiento media (MTR)

Como se ha dicho anteriormente, la tasa de rendimiento media (MTR) es el cociente entre la demanda cumplida y la demanda generada, como podemos ver en la siguiente fórmula:

$$MTR = \frac{\sum_{i=1}^n SD_i}{\sum_{i=1}^n GD_i} \quad (1)$$

#### Suma de la cola Kanban promedio (AAKQ)

$$AAKQ = \sum_{j=1}^m I_j \quad (2)$$

, donde  $I_j$  es el número promedio de Kanban esperando.

La letra  $j$  se refiere a cada estación del total de estaciones  $m$  en el sistema

Como es de suponer con lo dicho anteriormente, la tasa de rendimiento media debe ser maximizada y el total de la cola Kanban debe ser minimizado. Ya tenemos los dos criterios, por lo que ya podemos determinar la función objetivo.

Función de doble criterio

Como el número de tarjetas Kanban y también el tamaño de lote influyen en gran medida en esta expresión, y su naturaleza no está del todo definida, esta misma expresión AAKQ sufre un cambio para ser una función maximizada (TAAKQ).

$$TAAKQ = \frac{\sum_{j=1}^m IU_j - \sum_{j=1}^m I_j}{\sum_{j=1}^m IU_j - \sum_{j=1}^m IL_j} \quad (3)$$

$IU_j$ : número máximo de Kanban por cada estación

$IL_j$ : número mínimo de Kanban en la estación  $j$

Por lo tanto, al combinar ambas funciones, tanto la función MTR como la función TAAKQ nos queda la siguiente expresión, la función objetivo  $\zeta$ :

$$\zeta = \max [f(MTR, TAAKQ)] = \frac{\sum_{i=1}^n SD_i}{\sum_{i=1}^n GD_i} + \frac{\sum_{j=1}^m IU_j - \sum_{j=1}^m I_j}{\sum_{j=1}^m IU_j - \sum_{j=1}^m IL_j} \quad (4)$$

Esta función se va a ir utilizando para determinar la medida total del sistema.

## TÉCNICA DE BÚSQUEDA

Como se ha indicado anteriormente, las técnicas heurísticas no son las más utilizadas como herramientas para determinar el número de tarjetas Kanban. Sin embargo, ciertas heurísticas sí que nos pueden servir como solución del problema. Para el caso de técnicas de búsqueda, el recocido simulado se ha utilizado en otro tipo de campos, obteniendo buenos resultados, por lo que en este método la técnica del recocido simulado va a ser la utilizada.

## ALGORITMO UTILIZADO

Una vez dada una breve explicación acerca del recocido simulado, vamos a presentar el algoritmo de búsqueda utilizado, aunque la estructura mayormente utilizada para este tipo de algoritmos es la siguiente:

1. Generar una solución aleatoria.
2. Establecer una temperatura inicial  $T$  y un factor de enfriamiento  $\alpha$ .
3. Repetir hasta que se alcance un criterio de parada.
4. Devolver la mejor solución encontrada durante la ejecución.

La solución inicial aleatoria y la forma en que se generan las soluciones vecinas pueden variar según el problema específico que se esté abordando. La elección de la temperatura inicial y el factor de enfriamiento también es importante para el

desempeño del algoritmo. Un valor muy alto de la temperatura inicial puede hacer que el algoritmo se quede atascado en soluciones subóptimas, mientras que un valor muy bajo puede hacer que el algoritmo converja muy lentamente. El factor de enfriamiento  $\alpha$  debe ser elegido de manera que la temperatura disminuya lo suficientemente rápido para permitir que el algoritmo explore diferentes soluciones, pero no demasiado rápido para que no se pierdan soluciones potencialmente buenas.

En tal caso, el algoritmo utilizado es el siguiente:

- Paso 1: Se selecciona una semilla inicial  $S_0$ , un criterio de parada  $\pi$ , una temperatura inicial  $t_i$ , un factor de enfriamiento  $\alpha$ , y una temperatura final  $t_f$ . Tenemos que determinar  $f_0$ , que es el valor de la función objetivo en  $S_0$ , y actualizar  $f^*$ , el máximo global de la función objetivo y su correspondiente solución  $S^*$ .
- Paso 2: Una vez encontrada la solución  $S^*$ , tenemos que determinar el vecindario  $S_{n+1}$  usando perturbación y calcular  $f_{n+1}$ .
- Paso 3: Ahora el principal objetivo es encontrar nuevas soluciones, por lo que vamos a determinar lo siguiente:
  - Si  $(\delta = f_{n+1} - f) > 0$ , fijamos  $n+1$
  - Si no, entonces si  $r \leq e^{(-\delta/t_n)}$  donde  $r$  es un número aleatorio escogido entre los valores de 0 y 1, entonces también fijamos  $n+1$ .
- Paso 4: Actualizamos  $f^*$  y  $S^*$
- Paso 5: Si el criterio de parada es alcanzado, entonces ya vamos al último paso, si no, se reduce la temperatura con el factor de enfriamiento y se vuelve al paso 2.
- Paso 6: Ya tenemos la solución  $f^*$  y  $S^*$

## ELECCIÓN DE PARÁMETROS

Según varios estudios el valor de los parámetros más adecuado es el siguiente:

- Temperatura inicial ( $t_i$ ): 60
- Temperatura final ( $t_f$ ): 1
- Factor de enfriamiento ( $\alpha$ ): 0,8

En cuanto al número de iteraciones que se van a llevar a cabo tenemos contadores: el contador de iteraciones aceptadas y el contador total. Para una determinada temperatura, el número de iteraciones se dan por terminadas cuando el total de iteraciones aceptadas es igual al límite de aceptación ( $\beta$ ), o cuando el número total de iteraciones es igual a 4 veces  $\beta$ .

El algoritmo del recocido simulado se da por finalizado cuando se cumplen una de estas dos condiciones:

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

- La temperatura cae a un valor específico (tf)
- En el caso de que el contador de enfriamiento llegue a 5, es decir, cuando la tasa de iteraciones aceptadas / iteraciones totales es menor de 0,15 se incrementa el valor en 1. Una vez alcanza el valor de 5, el algoritmo se da por terminado.

Los parámetros que se han utilizado por (Shahabudeen, Gopinath, & Krishnaiah, 2002) se muestran a continuación:

System parameters

Part type	Processing times at workstations (time units)			Demand qty (units)	Demand inter arrival time (time units)
	WS1	WS2	WS3		
A	5 ± 1	4 ± 1	3 ± 2	200 ± 50	30 ± 20
B	3 ± 2	5 ± 2	4 ± 3	250 ± 40	40 ± 10

Tabla 1: Parámetros elegidos para el algoritmo. Fuente: (Shahabudeen, Gopinath, & Krishnaiah, 2002)

#### CONDICIONES DEL SISTEMA KANBAN

En esta investigación, las características del sistema son las siguientes:

- Se producen dos trabajos distintos
- Cada tipo de trabajo tiene su propia distribución de demanda entre llegadas
- Cada tipo de trabajo tiene su propia distribución de demanda
- Todos los trabajos deben ser procesados en todas las estaciones
- Los tiempos de procesado de cada trabajo son distintos
- Se utiliza un sistema FIFO
- Hay una cantidad infinita de materiales de entrada
- Todo Kanban independiente en la salida de una estación está inmediatamente disponible para la siguiente estación
- La distancia entre los centros de trabajo no es de interés
- Los recursos materiales no tienen ningún tipo de restricción.

Se trata de una línea de producción en la que tenemos tres centros de trabajo con dos productos distintos. Una vez llegan los componentes, se comprueba que están disponibles y en caso afirmativo, la producción comienza. En caso contrario se lanza un pedido para que pueda comenzar a funcionar la línea. De este modo, la producción en una etapa de la línea de producción solo puede ser completada una vez que la anterior ha finalizado.

Solamente una vez que ha finalizado el proceso de cada etapa, el Kanban del contenedor que ha sido vaciado se envía al inventario de salida de la estación precedente para que este contenedor pueda ser reemplazado por uno lleno, en el caso de haber disponibilidad.

En las siguientes tablas podemos ver los resultados que han dado lugar al experimento realizado por (Shahabudeen, Gopinath, & Krishnaiah, 2002).

Objective function values for different lot sizes (number of kanbans at each station = 5)

Lot size	MTR	TAAKQ	Total (Z)
20	36.48	55.79	92.27
30	27.112	56.82	83.93
40	44.88	55.29	100.17
50	43.88	55.74	99.62
60	31.38	51.95	83.33
70	40.62	50.48	91.10
80	52.68	51.38	104.06
90	49.00	53.02	102.02
100	57.08	51.09	108.17
110	41.46	56.34	97.80
120	47.13	54.60	101.73
130	37.18	53.25	90.43

Tabla 2: Resultados del algoritmo. Fuente: (Shahabudeen, Gopinath, & Krishnaiah, 2002)

En esta tabla podemos ver los resultados utilizando un total de 5 Kanban por cada estación en función del tamaño de lote. El total (Z) surge de la suma de la tasa de rendimiento acumulada y de la función de cola Kanban maximizada. Podemos ver que no hay grandes fluctuaciones, alcanzando un buen grado de estabilidad para un tamaño de lote entre 60 y 80.

## CONCLUSIONES

En este primer artículo que se ha presentado de (Shahabudeen, Gopinath, & Krishnaiah, 2002), se ha hecho un resumen de las ideas principales del autor y el procedimiento utilizado para la obtención del tamaño de lote óptimo en una línea de producción en la que se utiliza el sistema Kanban.

A partir de la aplicación del algoritmo metaheurístico del recocido simulado, se ha fijado un número de Kanban y se ha ido comprobando el comportamiento del modelo en función de cada tamaño de lote elegido por el autor.

### 3.1.2 ALGORITMO MEMÉTICO

Un algoritmo memético es una técnica de optimización inspirada en la teoría evolutiva y la palabra en inglés “meme”, que son ideas o comportamientos que se transmiten culturalmente de persona a persona. La idea detrás de los algoritmos meméticos es que las soluciones óptimas pueden ser encontradas mediante la combinación de los elementos más efectivos de soluciones previas.

En un algoritmo memético, la población de soluciones evoluciona a través de generaciones, y se aplican operadores evolutivos como la selección, la mutación y la recombinación para generar nuevas soluciones. Además, se aplican operadores de mejora local, que buscan mejorar las soluciones existentes a través de pequeñas modificaciones. (Priyom Dutta, 2022) da una idea clara con la siguiente ilustración:

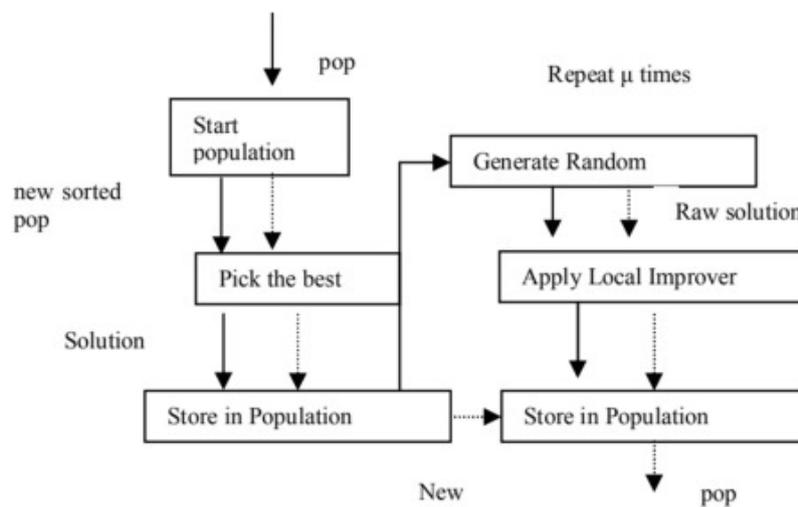


Ilustración 7: Pasos a seguir en un algoritmo memético. Fuente: (Priyom Dutta, 2022)

La principal diferencia entre los algoritmos genéticos y los algoritmos meméticos es que en los últimos se hace uso de la memoria a largo plazo para guiar la búsqueda de soluciones óptimas. En otras palabras, los algoritmos meméticos tratan de aprender de soluciones previas y de aplicar esa experiencia para mejorar la calidad de las soluciones actuales.

Los algoritmos meméticos han demostrado ser efectivos para resolver una amplia variedad de problemas de optimización en diferentes campos, como la ingeniería, la ciencia de la computación, la economía y la biología.

## MODELO

El objetivo principal de esta investigación de (Rabbani, Layegh, & Ebrahim, 2009) es evaluar las cadenas de suministro de más de una estación controladas por un sistema Kanban. En este sistema, la toma de decisiones está basada principalmente en dos aspectos: la determinación del número de Kanban y el tamaño de lote óptimo, como se ha visto en el caso anterior del recocido simulado, a partir de la investigación de (Rabbani, Layegh, & Ebrahim, 2009)

Una cadena de suministro es el conjunto de actividades y procesos que se llevan a cabo para producir y entregar un producto o servicio desde el proveedor de materias primas hasta el consumidor final. Incluye todos los pasos necesarios para convertir las materias primas en un producto final, incluyendo la planificación, la fabricación, el transporte, el almacenamiento y la distribución.

Su funcionamiento a largo plazo está asegurado si cumple con una producción de alta calidad, bajo coste y con entregas a tiempo, sin estar fuera de plazo. Tenemos dos tipos que con los siguientes:

- Cadenas de suministro de etapa única (SSSCS)
- Cadenas de suministro de etapa múltiple (MSSCS)

En el caso de tener dos plantas, se trata de una cadena de etapa única, pero en cuanto tenga más de dos, se trata de una cadena de suministro de etapa múltiple.

El sistema de cadena de suministro utilizado consta de proveedor, fábrica, mayorista y consumidor. Tanto el mayorista como el minorista vienen juntos en un subsistema.

La cadena está formada por  $n$  etapas de producción y cada planta dispone de dos almacenes. Uno está localizado antes de la planta (almacén A), con el objetivo de almacenar los distintos inputs, y el otro a la salida de la planta (almacén B), para almacenar todos los productos procesados en esa misma planta.

En el caso de que haya un almacén entre una supuesta planta  $i$  y otra planta  $i + 1$ , cuando la planta  $i + 1$  se dispone a utilizar un contenedor, se manda un Kanban de transporte y se coloca en el buzón Kanban. Más tarde, todos los Kanban de transporte recogidos del buzón se colocan en los contenedores vacíos que vuelven al almacén en cuestión, y así sucesivamente.

Las suposiciones que vamos a emplear en el modelo van a ser las siguientes:

- La tasa de demanda es menor que el cuello de botella. Además, la tasa de producción de cada planta va a ser mayor que la tasa de demanda, estando la escasez de productos impedida por decreto.
- Tenemos una demanda conocida
- En cada etapa el ritmo de producción es mayor que en la siguiente etapa.

En este problema, el principal objetivo es obtener una fórmula que minimice los costes totales en una cadena de suministro. El coste de cada planta es una función de transporte, manteniendo los costes de inventario.

Una función de transporte es una herramienta matemática que nos permite entender cómo se mueve o se transforma algo de una ubicación a otra, y se utiliza en diversas disciplinas para modelar y analizar una amplia gama de procesos. La notación utilizada para esta función es la siguiente:

- $i$ : subíndice para cada planta
- $j$ : subíndice para productos/bienes terminados
- $r$ : subíndice para la materia prima

Por otro lado, los parámetros utilizados son los que se nombran a continuación:

- $p_i$ : tasa de producción de cada planta  $i$  (desde  $i=1$  hasta  $N+1$ ); unidades/año
- $D$ : tasa de demanda; unidades/año
- $C$ : coste de un producto
- $H_r$ : Coste de mantener materia prima del inventario
- $H_{wi}$ : Coste de mantener el trabajo en curso del inventario en una etapa  $i$
- $H_f$ : Coste de mantenimiento del inventario de los productos terminados
- $A_r$ : Coste de puesta en marcha del pedido
- $A_{si}$ : Coste de puesta en marcha de la fabricación
- $A_{wi}$ : Coste de puesta en marcha del transporte en la etapa  $i$
- $A_f$ : coste de puesta en marcha en la estación  $N+1$
- $I(t)$ : Nivel de inventario
- $I_{ave}$ : Inventario promedio
- $T$ : tiempo de ciclo
- $TC_r$ : Coste de inventario de la materia prima
- $TC_{wi}$ : Coste de inventario del trabajo en curso
- $TC_f$ : Coste de inventario de los productos terminados
- $TC$ : Coste total de una cadena de suministro

Variables del sistema:

- $Q$ : Cantidad total de productos terminados producidos durante un período  $T$
- $Q_r$ : Cantidad ordenada, (unidades/orden)
- $Q_{wi}$ : Cantidad de trabajo en curso enviado (unidades/envío)
- $Q_f$ : Cantidad enviada de productos terminados
- $K_i$ : Número de envíos en la etapa  $i$
- $n_0$ : Número de orden del inventario de materia prima depositado

- $m_i$ : Número de envíos colocados durante el tiempo de funcionamiento de la producción en la etapa  $i$
- $n$ : Número de envíos colocados en la planta final, en la que tenemos los productos terminados.

#### Coste de inventario de la materia prima

El coste de la materia prima está formado por su coste de pedido y por el coste de mantenimiento. Entonces suponemos que la materia prima entra con una tasa de producción  $p_0$ , y la primera planta utiliza materia prima con una tasa de producción  $p_1$ . Su funcionamiento se puede apreciar gracias a la ilustración aportada por (Rabbani, Layegh, & Ebrahim, 2009).

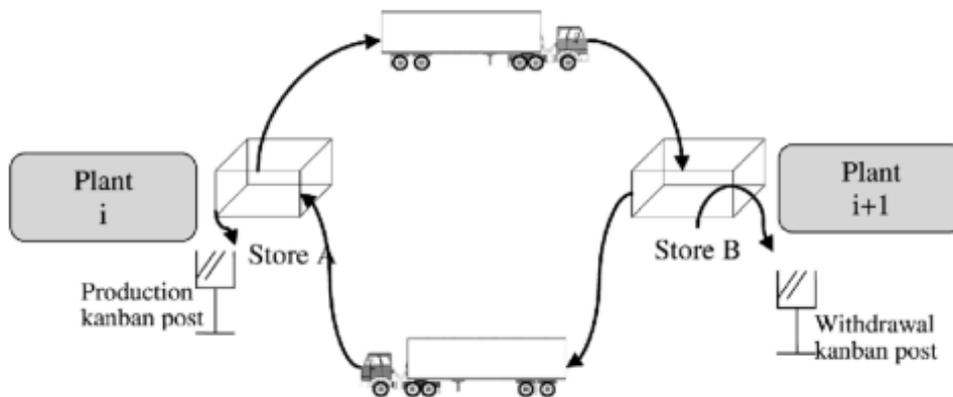


Ilustración 8: Esquema de funcionamiento de la línea de producción. Fuente: (Rabbani, Layegh, & Ebrahim, 2009)

#### Coste de inventario del trabajo en curso

Tiene en cuenta el coste de mantenimiento del inventario, el coste de transporte del contenedor y el coste de puesta en marcha para la producción. El coste de mantenimiento del inventario se da, tanto en los almacenes donde se lleva a cabo la producción, como en el almacén de la próxima planta a donde se lleva el contenedor.

Los parámetros para la solución del problema son los siguientes:

- $COI_{Ai}$ : Coste de mantenimiento de inventario del almacén A en la etapa  $i$
- $COI_{Bi}$ : Coste de mantenimiento de inventario del almacén B en la etapa  $i$
- $I_{Amax}$ : Máximo nivel de inventario del almacén A en la etapa  $i$
- $I_{Amin}$ : Mínimo nivel de inventario del almacén A en la etapa  $i$
- $I_{Aiave}$ : Inventario promedio del almacén A en la etapa  $i$
- $I_{Biave}$ : Inventario promedio del almacén B en la etapa  $i$

- $COI_i$ : Coste de mantenimiento de inventario en la etapa  $i$

Como es lógico, el coste de mantenimiento de inventario en la etapa  $i$  es la suma del inventario de la etapa  $i$  correspondiente al almacén A más el correspondiente al almacén B:

$$COI_i = COI_{Ai} + COI_{Bi}$$

#### PRIMER TEOREMA

El TC óptimo, es decir, el coste total, es la solución ideal del problema. El óptimo más realista que podemos llegar a desarrollar siempre es algo peor que la solución ideal. Este óptimo se puede utilizar como un límite inferior.

Esto se prueba más adelante calculando las derivadas parciales correspondientes de la ecuación principal dada, TC, con sus respectivos  $x_0, x_1, \dots, x_N$ , lo que nos lleva a las ecuaciones las ecuaciones finales que se introducirán en el algoritmo.

#### RESOLUCIÓN DEL PROBLEMA

Los problemas de optimización combinatoria tienen gran importancia en muchos ámbitos de la ciencia y muchos de ellos tienen una complejidad enorme, por lo que para ello se utilizan algunos métodos exactos como el B&B (Branch and Bound)

Como ya es sabido, estos problemas son muy complejos, lo que hace muy complicado encontrar una herramienta que nos pueda proporcionar una solución óptima. En cambio, tenemos algunos que a pesar de que no nos proporcionan esa mejor solución que estamos encontrando, nos acercan a una solución vecina que se aproxima de una manera bastante precisa.

Como se ha comentado en la introducción a esta metodología de resolución del problema, los métodos heurísticos son métodos de búsqueda que nos proporcionan una solución real en un tiempo relativamente corto.

La diferencia que hay entre los métodos metaheurísticos y los heurísticos, es que los metaheurísticos tienen la ventaja de tener un marco de trabajo específico, y pueden combinarse con otras heurísticas. Este es el caso del algoritmo memético, que se ha ido desarrollando a lo largo de este artículo.

#### MÉTODO PROPUESTO: INICIALIZACIÓN

El primer paso que se da es generar una solución inicial y fijar una población inicial. A partir de ello, el número de  $n$  soluciones iniciales está generado a partir de las soluciones aproximadas que nos da el primer teorema. En este artículo, se han generado las soluciones iniciales calculando la parte entera más próxima, tanto inferior como superior ("floor and ceiling functions") de los genes de la solución óptima no factible del primer teorema.

Se comienza escogiendo un primer gen y remplazamos o bien su parte entera inferior más cercana o la parte entera superior más cercana, escogido esto de manera aleatoria, e ir haciendo este paso respectivamente con todos los otros genes.

Ahora ya se ha generado una solución factible, y se debe ir repitiendo esto más veces para obtener nuevas soluciones iniciales, dando lugar al vector solución  $X = (1, \dots, 1)$  y a  $Y$ , que es un vector cuyos genes son suficientemente más grandes que los genes de la solución óptima no factible para la colección de las soluciones generadas. Todas ellas se añaden a un conjunto de soluciones denominado  $SS$ .

### BÚSQUEDA LOCAL

El método de búsqueda local es una técnica utilizada en la optimización que busca una solución óptima a un problema dado, explorando iterativamente soluciones cercanas a una solución inicial.

En general, el método de búsqueda local comienza con una solución inicial, como es el caso de este problema, que puede ser generada aleatoriamente o a través de un proceso de selección previo. A continuación, se evalúa la calidad de la solución inicial y se buscan soluciones cercanas a ella que puedan mejorar la calidad de la solución.

La búsqueda de soluciones cercanas se realiza moviéndose a lo largo de una dirección de búsqueda y evaluando la calidad de la solución en cada paso. Si se encuentra una solución que mejora la calidad de la solución actual, se toma como la nueva solución actual y se repite el proceso. Si no se encuentra una solución mejor, se detiene la búsqueda y se devuelve la solución actual.

El método de búsqueda local es útil cuando el espacio de búsqueda es demasiado grande para explorar todas las posibles soluciones, o cuando no se dispone de información completa sobre el problema. Sin embargo, es importante destacar que el método de búsqueda local puede quedar atrapado en un óptimo local, lo que significa que puede encontrar una solución que es la mejor en su vecindario, pero no es la mejor en el espacio de búsqueda global.

Se ha escogido una búsqueda local con más de un punto de comienzo, concretamente tres. Esto tiene sentido ya que tenemos que obtener dos soluciones a partir de la solución principal. Para obtener estas dos soluciones de la solución principal tenemos primeramente que añadir una unidad a cada componente del vector de la solución principal. Una de estas dos soluciones se ha obtenido y construimos la segunda solución de una manera similar, pero esta vez tenemos que disminuir un elemento para cada componente.

En el caso de la segunda solución hay que tener cuidado ya que todas las componentes deben ser positivas. Usando el algoritmo de búsqueda local, se generan soluciones aleatorias de cada uno de estos tres puntos.

Este proceso se alarga en el tiempo hasta que el porcentaje de disminución de la función de fitness (en nuestro caso es el TC óptimo, referido al coste total) tiene un resultado negativo para un número  $\mu$  de veces.

Una vez que ya tenemos esas soluciones óptimas a partir de la búsqueda local, actualizamos la colección de soluciones (SS). Su estructura se puede apreciar en la siguiente imagen de (Rabbani, Layegh, & Ebrahim, 2009).

```
Procedure MA;  
  
Begin  
  For j := 1 to popsize do  
    i := generate solution();  
    i := local search(i);  
    Add individual i to p;  
  End for;  
  Repeat  
    For i := 1 to # recombinations do  
      Select two parents  $i_a, i_b \in p$  randomly  
       $i_c :=$  recombine ( $i_a, i_b$ );  
       $i_c :=$  localsearch ( $i_c$ );  
      Add individual  $i_c$  to p;  
    End for;  
    For i := 1 to # mutation do  
      Select an individual  $i \in p$  randomly;  
       $i_m :=$  mutate(i);  
       $i_m :=$  local search ( $i_m$ );  
      Add individual  $i_m$  to p;  
    End for;  
    p := select(p);  
    If p converged then p := mutate and LS(p);  
  Until terminate= true;  
End;
```

*Ilustración 9: Estructura principal del algoritmo memético. Fuente: (Rabbani, Layegh, & Ebrahim, 2009)*

## RECOMBINACIÓN

En los algoritmos genéticos, la recombinación (también conocida como cruce o crossover) es un operador que se utiliza para crear nuevas soluciones a partir de combinaciones de soluciones existentes.

La recombinación se basa en el principio de que, al combinar características de soluciones existentes, se pueden crear nuevas soluciones con mejores características que las soluciones iniciales. En el proceso de recombinación, se seleccionan dos soluciones existentes de la población de soluciones y se intercambian partes de ellas para crear dos soluciones nuevas.

La recombinación se aplica generalmente después de la selección de las soluciones más aptas en la población, y antes de la aplicación de la mutación. La probabilidad de recombinación se establece generalmente entre el 50% y el 95%.

En este problema se utiliza un operador binario para combinar la pareja de soluciones de la que se ha hablado anteriormente, y encontrar la mejor solución. En este paso, el objetivo principal es mejorar soluciones peores, es decir, las que tienen una mayor variación de la función de fitness, por lo que una combinación de estas dos soluciones es medida.

También se hace evaluando las mejores y peores soluciones, y evaluando también su función de fitness. Para ello tenemos que ordenar todas las soluciones existentes en el conjunto de soluciones SS y escoger las dos soluciones peores como las peores soluciones, y otra vez este paso con la peor y la mejor solución.

Se escogen dos soluciones W1 y W2 con un valor de lambda de 0,5. El funcionamiento de la recombinación propuesto por (Rabbani, Layegh, & Ebrahim, 2009) es el siguiente:

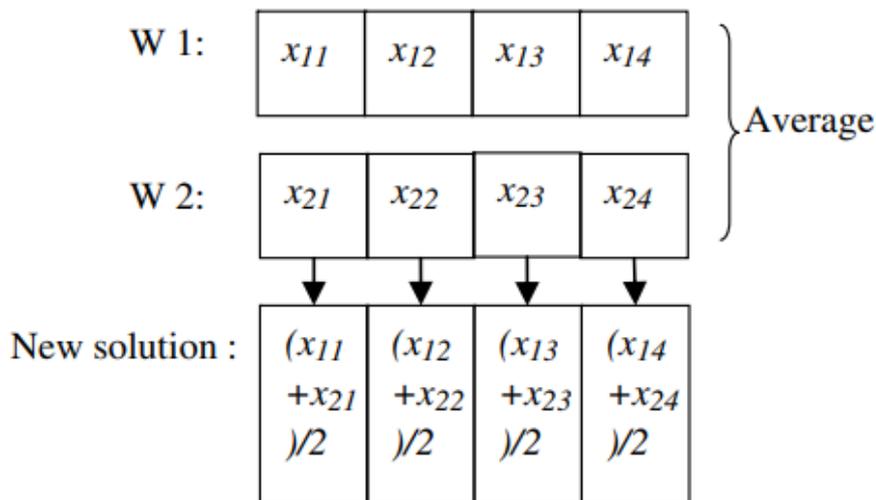


Ilustración 10: Esquema del funcionamiento de la recombinación. Fuente: (Rabbani, Layegh, & Ebrahim, 2009)

## MUTACIÓN

La mutación es un operador que introduce variabilidad en la población de soluciones y ayuda a evitar que se llegue de manera muy temprana a una solución óptima.

La mutación es un proceso que se asemeja a las mutaciones genéticas en la naturaleza, donde se produce un cambio aleatorio en el material genético de un organismo. En los algoritmos genéticos, la mutación implica hacer pequeños cambios aleatorios en las soluciones de la población, como cambiar un bit en una cadena binaria o añadir o eliminar una variable en una representación de una solución.

La mutación se aplica normalmente con una probabilidad baja, típicamente alrededor del 1% al 5%, para que no tenga un impacto excesivo en la población de soluciones. Además, la mutación suele ser complementaria a otros operadores de los algoritmos genéticos, como la selección y la recombinación, para explorar eficazmente el espacio de búsqueda de soluciones.

En este artículo se usa como datos de entrada las mejores y peores soluciones generadas. Se calculan las primeras derivadas del TC respetando las variables de decisión para cada entrada.

Se sabe que la primera derivada de una función como el TC siempre es positiva, independientemente de las variables de entrada, por lo que un pequeño aumento en una variable podría llevarnos a un incremento en el valor de la función.

Esto sucede a la inversa en el caso de que la derivada sea negativa: una pequeña disminución en esa variable podría llevar a un decremento del valor de la función.

Estas primeras derivadas se evalúan para cada uno de los componentes del vector de mejores y peores soluciones. El componente con mayor impacto, es decir, el que tenga el mayor valor en la derivada en la función TC es considerado, y en el caso de ser positivo su valor se reduce en una unidad. En el caso de que sea negativo sucede lo contrario, se aumenta su valor en una unidad.

Una vez que tenemos estas soluciones solamente es necesario evaluar su valor en la fórmula del TC antes de añadir las al conjunto de soluciones (SS).

### EJEMPLO

En este artículo se presenta una resolución al problema que es la que viene a continuación, primero mostrando los parámetros utilizados por (Rabbani, Layegh, & Ebrahim, 2009):

Holding cost	Setup cost	Shipping cost	Production rate
$P_0 = 7000$			
$P_1 = 6500$	$A_{s1} = 300$	$A_r = 110$	$H_r = 45$
$P_2 = 6000$	$A_{s2} = 250$	$A_{w1} = 100$	$H_{w1} = 30$
$P_3 = 5500$	$A_{s3} = 300$	$A_{w2} = 80$	$H_{w2} = 45$
$P_4 = D = 5000$		$A_{w3} = 120$	$H_{w3} = 25$

Tabla 3: Parámetros del sistema utilizados. Fuente: (Rabbani, Layegh, & Ebrahim, 2009)

Parámetros utilizados:

- Ritmo de producción: Uniforme (5000,7500)
- Coste de envío: Uniforme (200,400)
- Coste de preparación: Uniforme (70,150)
- Coste de inventario: Uniforme (20,55)

Se ha hecho una aproximación para la fórmula del TC óptimo para así tener un menor coste computacional, y las soluciones aportadas son las siguientes:

1. La solución ideal no factible se ha calculado y los resultados son:

$$X_0 = 1.807; \quad x_1 = 5,183; \quad x_2 = 7,4; \quad x_3 = 4,71; \quad TC \text{ (ideal)} = 25774,873$$

2. La solución por el algoritmo propuesto es la siguiente:

$$X_0 = 2; \quad x_1 = 5; \quad x_2 = 8; \quad x_3 = 5; \quad TC \text{ (MO)} = 25802,59$$

Donde MO es memetic output, es decir, los resultados de salida del algoritmo memético

A continuación, se introducen dos parámetros con el objetivo de evaluar la calidad de los resultados de salida obtenidos a partir del algoritmo propuesto.

En la tabla final de soluciones tenemos el % relativo de MO respecto a la solución ideal y el % relativo de desviación de MO respecto B&B.

3. La solución proporcionada por B&B

$$X_0 = 2; \quad x_1 = 5; \quad x_2 = 8; \quad x_3 = 5; \quad TC \text{ (MO)} = 25802,60$$

Estos resultados que se acaban de mostrar son para una cadena de suministro con tres etapas distintas. La diferencia de resultados entre B&B y el algoritmo memético es casi nula, pero esa pequeña diferencia se debe al redondeo en los resultados de salida.

El experimento se repite para cadenas de suministro con mayor número de etapas, hasta llegar a un total de 30, siendo la diferencia entre los dos métodos prácticamente nula, mostrándose los siguientes resultados por (Rabbani, Layegh, & Ebrahim, 2009):

Number of plants	Ideal solutions	Memetic outputs (MO)	B&B outputs	% of MO deviation from ideal solution	% of MO deviation from B&B
N = 3	25774.87	25802.60	25802.6	0.10756	0
N = 4	29330.33	29355.08	29355.08	0.08439	0
N = 5	34125.74	34172.61	34172.61	0.13735	0
N = 6	38627.23	38665.64	38665.64	0.09944	0
N = 7	43335.03	43368.28	43368.28	0.07672	0
N = 8	50142.95	50193.07	50169.6	0.09996	0.04677
N = 9	56264.61	56286.75	56286.75	0.03936	0
N = 10	58107.89	58132.31	58132.31	0.04204	0
N = 16	113180.44	113217.05	113214.6	0.03235	2.16E-03
N = 30	144863.05	144958.31	144933.3	0.06576	0.01726

Tabla 4: Tabla de resultados. Fuente: (Rabbani, Layegh, & Ebrahim, 2009)

## CONCLUSIONES

En este artículo se parte de una planta de producción que implementa un sistema Kanban, donde el principal problema es calcular el mínimo coste total del sistema.

Se ha podido comprobar la eficiencia de los algoritmos genéticos como herramienta de apoyo para la obtención de los resultados de salida del algoritmo memético y su comparación con los resultados de una solución ideal y de un método exacto.

Una vez se han comparado las soluciones tanto del algoritmo memético (MA), como de B&B, y de las soluciones ideales, podemos llegar a la conclusión de que las soluciones propuestas por algoritmos meméticos tienen una gran fiabilidad como se puede observar en los datos de la tabla anterior, desviándose muy poco o prácticamente nada respecto a la solución ideal.

En definitiva, este método puede ser de gran utilidad a la hora de obtener el número óptimo de tarjetas Kanban y el tamaño de lote en problemas de grandes dimensiones.

### 3.1.3 ALGORITMO GENÉTICO MULTI OBJETIVO

En este artículo de (Hou & Hu, 2011) se utilizan algoritmos genéticos multi objetivo para determinar el óptimo de Pareto para el número de Kanban y el tamaño de lote. Estos resultados se aplican en una compañía orientada al Just In Time para demostrar su viabilidad. Se lleva a cabo también un modelo de simulación para simular el todo el proceso productivo.

Luego con un modelo experimental es aplicado para comprobar los resultados productivos. A partir de ello, se construye un modelo de regresión para representar la relación entre el número de Kanban de las diferentes etapas de producción o del desempeño de la producción. Los modelos de regresión son utilizados en algoritmos genéticos para generar la función de los cromosomas.

El algoritmo genético multiobjetivo propuesto (MOGA) utiliza la Función de fitness de escala independiente basada en la Distribución de Pareto Generalizada (GPSIFF) como una función de fitness para evaluar los distintos objetivos para los cromosomas y para finalmente determinar el óptimo de Pareto que nos permitirá determinar el número y tamaño de los Kanban correspondientes.

## ALGORITMOS EVOLUTIVOS MULTI OBJETIVO

Los problemas en los que hay objetivos que entran en conflicto son muy comunes a la hora de aplicarse en el mundo real. Este tipo de problemas son denominados multi objetivo o multi criterio.

Los algoritmos evolutivos multiobjetivo (MOEA's, por sus siglas en inglés) son un tipo de algoritmo de optimización que se utiliza para resolver problemas que involucran múltiples objetivos o criterios de optimización. Estos algoritmos están basados en la idea de la selección natural y la evolución biológica.

En un problema multiobjetivo, se busca optimizar varios objetivos simultáneamente, lo que puede llevar a soluciones que no son únicas y a menudo son contradictorias entre sí. Los MOEA's abordan este problema generando una población de soluciones candidatas y aplicando operaciones de selección, reproducción y mutación para generar nuevas soluciones y mejorar la calidad de la población existente.

La principal característica de los MOEA's es que buscan encontrar un conjunto de soluciones óptimas o no dominadas, es decir, soluciones que no pueden ser mejoradas en un objetivo sin empeorar en otro objetivo. Este conjunto de soluciones óptimas se conoce como la frontera de Pareto o conjunto de Pareto.

La mayoría de este tipo de problemas usa el concepto de dominancia para encontrar las soluciones óptimas de Pareto. La solución dominada sigue la regla que viene a continuación:

Se dice que una solución  $x_{(1)}$  domina a otra solución  $x_{(2)}$  cuando ambas condiciones siguientes se cumplen:

- Primera condición: la solución  $x_{(1)}$  no es peor que  $x_{(2)}$  en todos los objetivos
- Segunda condición: la solución  $x_{(1)}$  es estrictamente mejor que  $x_{(2)}$  en al menos uno de los objetivos.

En cuanto una de estas dos condiciones no se cumple, la solución  $x_{(1)}$  ya no domina a la solución  $x_{(2)}$ .

En el caso de tener una función de doble objetivo, en el que uno de ellos sea maximizar y el otro minimizar, y suponiendo que ambos tienen la misma importancia a la hora de tomar una decisión, y una vez encontramos varias soluciones factibles, podemos ir observando que unas soluciones dominan a otras en ambos objetivos.

Usando el concepto de dominancia para comparar las soluciones en funciones de objetivo múltiple, la mayoría de los algoritmos evolutivos multi-objetivo (MOEA's) determinan que las soluciones dominadas también se encuentran dentro de la solución óptima de Pareto.

Una manera de determinar numéricamente la solución óptima de Pareto es con la llamada función de fitness de escala independiente basada en la Distribución de Pareto Generalizada, que es una función utilizada en la optimización multiobjetivo para evaluar el rendimiento de una solución en un espacio de búsqueda de múltiples objetivos.

Esta función de fitness utiliza la distribución de Pareto generalizada para transformar los valores objetivos en una escala independiente que permite comparar soluciones que tienen diferentes escalas y unidades de medida. La distribución de Pareto generalizada se ajusta a los valores objetivos de las soluciones y proporciona una medida de la calidad de la solución en términos de la probabilidad de que la solución sea dominada por otras soluciones.

La ventaja de esta función de aptitud es que permite evaluar soluciones que tienen diferentes escalas y unidades de medida, lo que facilita la comparación de soluciones y la selección de las mejores soluciones en términos de múltiples objetivos. Además, la función de aptitud es fácil de implementar y se puede utilizar en diferentes tipos de problemas de optimización multiobjetivo.

En este artículo de (Hou & Hu, 2011) la principal idea es evaluar la dominancia de cada solución utilizando lo que se dice en inglés “score function”, que es una función matemática que asigna un valor numérico o un marcador (score) para un conjunto de datos o para un modelo basado en algún criterio o conjunto de criterios.

En el caso que nos incumbe, que es la optimización, una “score function” se utiliza para evaluar la calidad de una solución en un problema de optimización. Esa función mide cómo de buena es esta función en base a la satisfacción de restricciones y objetivos del problema de optimización. Su función es, además, servir como guía para la búsqueda de mejores soluciones.

Para este problema, el cálculo de la “score function” se hace de acorde a los siguientes elementos:

$$\text{Score}(x) = p - q + c;$$

- $p$  es el número de soluciones dominadas por  $x$
- $q$  es el número de soluciones que dominan a  $x$
- $c$  es una constante escalar que hace que la función de fitness tenga un valor positivo.

Una vez calculamos todos los valores de la función score para todas las soluciones posibles, las soluciones que tengan un mayor valor pueden ser encontradas.

## SISTEMA PROPUESTO

Este apartado corresponde al modelo de simulación que se ha nombrado anteriormente, para simular un sistema productivo basado en Just In Time. A continuación, un diseño experimental para un número distinto de Kanban y de tamaños de lote para las diferentes etapas de producción es aplicado para evaluar su rendimiento.

Después de haber obtenido esos resultados, se realizan modelos de regresión para representar las relaciones entre el número de Kanban y el tamaño de Kanban para cada uno de los rendimientos de las fases de producción.

## MODELO DE SIMULACIÓN

En este modelo se utiliza la herramienta para simulación de procesos llamada SimProcess para la producción de techo solar para una línea de producción de automóviles para el caso de que la compañía utilice el sistema Just In Time.

En este caso la compañía utiliza un sistema Kanban de doble tarjeta para las partes ya ensambladas, y un sistema de tarjeta único para el trabajo en curso y las piezas del proveedor.

El cliente solicita un plan mensual de producción y una cadena de suministro para clientes mediante un sistema Kanban, de modo que proporcione las piezas terminadas al consumidor tres veces al día, y requiera de planes de compra de materiales mensualmente a sus proveedores.

Una vez las piezas terminadas son desplazadas de la zona de envíos, los Kanban de ensamblaje K3 son movidos del contenedor a la estación de ensamblaje que produce los techos solares siguiendo siempre el número Kanban K3. Estos Kanban se unirán de nuevo al contenedor de partes terminadas.

El mapa del flujo de valor de la línea de producción Kanban realizado por (Hou & Hu, 2011) es el que viene a continuación:

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

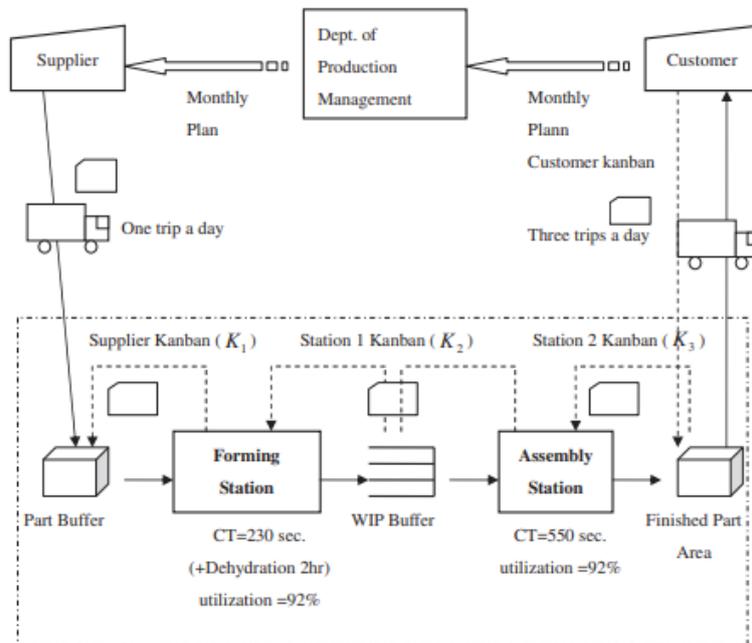


Ilustración 11: Value Stream mapping para la producción del techo solar. Fuente: (Hou & Hu, 2011)

Para ensamblar el techo solar, el operador en la estación de ensamblaje tiene que ir a la zona de trabajo en curso (WIP) para retirar las partes que están semiterminadas. A continuación, debe retirar el Kanban  $K_2$  del contenedor, ponerlo en una caja Kanban, y mover las partes de vuelta a la estación de ensamblaje.

En la estación de formación de piezas se deben producir piezas de acuerdo con el número Kanban  $K_2$ , y el operador tiene que ir a la zona de suministro para retirar las partes ya suministradas. El operario tiene que despegar el Kanban  $K_1$  del contenedor, ponerlo en una caja Kanban, y mover las partes de vuelta a la estación de formación.

El Kanban  $K_1$  tiene que ser mandado de vuelta al proveedor que transporta las piezas para la compañía una vez al día acorde al número de Kanban  $K_1$ .

La compañía en cuestión fija de manera experimental  $K_1$  en 4,  $K_2$  en 4 y  $K_3$  en 3, y el tamaño de cada Kanban en 4. El modelo de simulación realizado por (Hou & Hu, 2011) es el siguiente:



estadísticas determinadas para poder obtener, en este caso, el número óptimo de tarjetas Kanban y de tamaño de lote.

Al ser utilizadas dos medidas de rendimiento, se necesita construir dos modelos de regresión distintos, usándose normalmente modelos lineales y cuadráticos. En un primer momento se utilizará el modelo lineal para cuadrar el resultado de la simulación. Si la fitness o aptitud no es la deseada, se echará mano del modelo cuadrático.

En todo caso, los dos modelos lineales propuestos por (Hou & Hu, 2011) para la tasa de rendimiento acumulada y para el trabajo en curso, que se denominan por  $Z_{TR}$  y  $Z_{WIP}$ .

## APROXIMACIÓN DE ALGORITMOS GENÉTICOS MULTI OBJETIVO (MOGA) PROPUESTA

### Primer paso: Codificación del cromosoma

Se trata de una cadena binaria, en la que cada variable de decisión se codifica con su correspondiente valor binario con sus cuatro dígitos correspondientes. Por ejemplo, para el caso  $K1=4$ ,  $K2=4$ ,  $K3=3$ ,  $B=7$ , el cromosoma se forma de la siguiente manera:

- $K1=4$  en binario: 0100
- $K2=4$  en binario: 0100
- $K3=3$  en binario: 0011
- $B=7$  en binario: 0111

Juntando toda la cadena nos queda: 0100010000110111

### Segundo paso: Inicialización de la población

Los cromosomas son generados de manera totalmente aleatoria, y el tamaño de la población del cromosoma se denota como  $N_{pop}$ . Los conjuntos elitista  $E$  y el conjunto elitista temporal  $E'$  comienzan estando vacíos.

### Tercer paso: calcular el valor de la fitness para cada cromosoma

Los modelos de regresión  $Z_{TR}$  y  $Z_{WIP}$  nos sirven para calcular el valor de las funciones objetivo para cada cromosoma y la función score para cada cromosoma  $x$  es calculado utilizando su propia fórmula.

El valor del score( $x$ ) se genera usando la función de fitness de escala independiente basada en la Distribución de Pareto Generalizada (GPSIFF), que tiene en cuenta los valores de la fitness en el espacio de Pareto tanto para las soluciones dominadas como para las no dominadas.

Las soluciones no dominadas serán copiadas a un conjunto elitista temporal E' para actualizar el conjunto elitista E

#### Cuarto paso: Actualizar el conjunto elitista

Como primera acción, las soluciones no dominadas del conjunto elitista temporal E' se ponen en el conjunto elitista E. Por lo tanto, el conjunto E' se reinicia para que vuelva a estar vacío.

A continuación, las soluciones del conjunto E son evaluadas de nuevo por su no dominancia, y las soluciones dominadas son retiradas del conjunto elitista. Si las soluciones no dominadas NE son mayores que la capacidad máxima del conjunto Nmax, entonces algunas de las soluciones no dominadas tendrán que ser seleccionadas de manera aleatoria y posteriormente ser eliminadas del conjunto elitista.

#### Quinto paso: Selección

En esta operación se seleccionan los cromosomas para dar lugar al cruce, que es la etapa que viene en el siguiente paso. Cuanta mayor fitness tenga un cromosoma, mayor probabilidad tendrá de ser elegido.

#### Sexto paso: Cruce

Para cada par de cromosomas seleccionado, se lleva a cabo el cruce para que aparezcan nuevos "hijos", con una probabilidad Pc. Entonces un número aleatorio Rc se genera. Si  $Rc \leq Pc$ , entonces el cruce se realiza. En caso contrario, se va al siguiente paso, que es el séptimo.

#### Séptimo paso: Mutación

Existe una probabilidad de mutación Pm para engendrar nuevos cromosomas. Generamos otra vez un número aleatorio, esta vez denominado Rm. Si  $Rm \leq Pm$ , tienen lugar la mutación, y se cambian los 0's por 1's y viceversa. Usando ambas operaciones de cruce y mutación, una nueva población se genera, y los cromosomas de los padres son sustituidos por los de los hijos.

#### Octavo paso: Prueba final

Se utiliza un número Gmax para el criterio de parada. Si no se satisface una condición especificada de parada previamente, se vuelve al tercer paso. De caso de sí cumplirse, el algoritmo finalizará y el óptimo de Pareto será encontrado dentro de la población y del conjunto elitista E.

## RESULTADOS

En este artículo se utiliza como bien se ha dicho antes, la herramienta SimProcess para construir la simulación del proceso, que consiste en la compañía en cuestión, un proveedor y un cliente.

Los datos que se utilizan son los siguientes:

- Demanda mensual por parte del cliente de 680 unidades.
- Las demandas mensuales son niveladas para cada día y cada envío, habiendo tres envíos diarios, con un intervalo de dos horas y media.
- El tiempo de ciclo en la estación de formación es de 230 segundos, y el trabajo en curso (WIP) tiene que ser secado durante dos horas antes de que pueda ser utilizado en la estación de ensamblaje, donde el tiempo de ciclo es de 550 segundos.
- La tasa de menor calidad, o de defectos se fija en un 1%.
- La utilización de las máquinas de cada estación es de alrededor un 92%.
- Hay ocho horas de trabajo durante los 19 días de trabajo del mes.

El tiempo de simulación es de esos 19 días utilizando un tamaño de Kanban de 4, y números Kanban de  $K_1=4$ ,  $K_2=4$ ,  $K_3=3$ .

En este artículo (Hou & Hu, 2011) obtienen las siguientes soluciones:

No.	Chromosome	$(K_1, K_2, K_3, B)$	TR	WIP	Improvement (TR, WIP)
1	0010 0100 0100 0001	(2, 4, 4, 1)	15.947	3.958	(-66.7%, 87.9%)
2	0011 0111 0111 0001	(3, 7, 7, 1)	27.842	5.937	(-41.9%, 81.9%)
3	0100 1001 1001 0001	(4, 9, 9, 1)	35.842	7.917	(-25.2%, 75.9%)
4	0011 0101 0101 0010	(3, 5, 5, 2)	39.579	11.875	(-17.5%, 63.8%)
5	0011 0110 0110 0010	(3, 6, 6, 2)	45.684	12.056	(-4.7%, 63.2%)
6	0011 0111 0111 0010	(3, 7, 7, 2)	51.263	26.390	(6.9%, 19.5%)
7	0010 0100 0101 0100	(2, 4, 5, 4)	51.632	26.469	(7.7%, 19.3%)
8	0111 1000 1000 0010	(7, 8, 8, 2)	51.684	28.725	(7.8%, 12.4%)
9	0101 0110 0111 0011	(5, 6, 7, 3)	51.789	31.004	(8.0%, 5.5%)
10	0100 1000 1000 0100	(4, 8, 8, 4)	51.895	39.971	(8.2%, -21.9%)
	Current situation	(4, 4, 3, 4)	47.947	32.796	

Tabla 5: Soluciones de los óptimos de Pareto. Fuente: (Hou & Hu, 2011)

En esta figura tenemos sombreadas las mejores soluciones, donde podemos ver que el TR, que marca la eficiencia de la línea, tiene un mayor valor.

Se aplica primero un modelo lineal, pero al ser el  $R^2$  muy pequeño, se opta por el modelo cuadrático.

Los valores de  $R^2$  para el modelo cuadrático mejoran en gran medida a los del modelo lineal, siendo muy próximos a 1, concretamente 0,919 para el TR y 0,998 para el WIP, por lo que tienen un error muy pequeño. Por lo tanto, este modelo se puede utilizar para tamaños de lote dentro del intervalo 0-15.

Las soluciones proporcionadas por el óptimo de Pareto deberían ser válidas en la práctica.

## CONCLUSIONES

En resumen, en este artículo se encuentran que, dentro de los óptimos de Pareto, algunos tienen mayor rendimiento para el caso de TR y WIP, que el Kanban utilizado en la compañía en cuestión. Los resultados que se muestran otorgan una buena viabilidad al modelo.

Aunque el modelo se ha diseñado para un sistema Kanban tradicional, también podría aplicarse a un sistema Kanban flexible.

## 3.2 MÉTODOS MATEMÁTICOS

En este apartado los métodos matemáticos que me han parecido más interesantes para el cálculo de tarjetas Kanban.

Un método matemático es una técnica o procedimiento que se utiliza para resolver un problema matemático o para analizar un sistema o fenómeno utilizando conceptos matemáticos. Los métodos matemáticos se utilizan en diversas áreas, como la física, la ingeniería, la economía, la estadística, entre otras.

Cada método matemático tiene sus propias técnicas y herramientas específicas para resolver problemas y analizar sistemas. La elección del método adecuado depende del problema que se esté tratando y de los datos disponibles.

Entre los artículos que se van a presentar están los siguientes:

- Dimensionamiento del búfer para un sistema Heijunka Kanban
- Programación dinámica para una producción de línea de un producto con varias etapas controlada por Kanban

### 3.2.1 DIMENSIONAMIENTO DEL BÚFER PARA UN SISTEMA HEIJUNKA KANBAN

En este artículo de (Matzka, Di Mascolo, & Furmans, 2012), para el caso de una línea de producción en un sistema que aplica la metodología Kanban, el proveedor debe adaptar lo que produce a las órdenes del consumidor, y producirlo según al tiempo de ciclo (en inglés, takt time).

El tiempo de ciclo es usado para sincronizar el ritmo de producción con el ritmo de ventas, y se calcula dividiendo el tiempo disponible de trabajo diario entre la tasa de demanda diaria del cliente:

$$\text{takt time} = \frac{\text{tiempo diario disponible}}{\text{tasa de demanda diaria del cliente}} \quad (1)$$

Esta fórmula nos da un intervalo de tiempo en el que cada parte debe ser producida. Para ello se introduce el concepto de heijunka, que no solo persigue nivelar la producción, sino también nivelar el mix de productos utilizando la misma secuencia de productos para cada ciclo de producción.

Un cálculo del heijunka define una secuencia repetitiva de producción y los tamaños de lote para un intervalo programado, llamado EPEI (en inglés “each part each Interval”).

Nivelando el mix de producción y haciendo cambios entre productos de un modo más habitual, el proveedor está más capacitado para responder a los distintos requerimientos del consumidor, consiguiendo tener a su vez un pequeño inventario de bienes terminados.

La diferencia entre distintos modelos de producción requiere que cada cambio entre ítems sea rápido y lo más simple posible. Para ello heijunka dicta secuencias de producción de lotes pequeños, lo que sería muy ineficiente sin puestas en marcha ágiles.

El principal objetivo de Heijunka es proveer a uno o más consumidores procesos con un flujo de constante de pequeños lotes, generando a su vez una demanda constante en procesos aguas arriba.

#### DESCRIPCIÓN DEL SISTEMA

Tenemos un sistema de doble tarjeta, es decir, que distingue entre los Kanban de producción y los Kanban de transporte. Tiene simplemente una etapa en la que hay una máquina. Se producen varios productos en dicha máquina y tanto el volumen como el mix de producción son nivelados por un tablero heijunka.

El cliente utiliza Kanban para retirar productos terminados del buffer de salida. La cantidad recogida tiene mucha variabilidad, desde 0 hasta el número de Kanban de transporte en el bucle entre el cliente y el proveedor.

En la siguiente ilustración se va a mostrar el funcionamiento del tablero Heijunka propuesto por (Matzka, Di Mascolo, & Furmans, 2012):

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

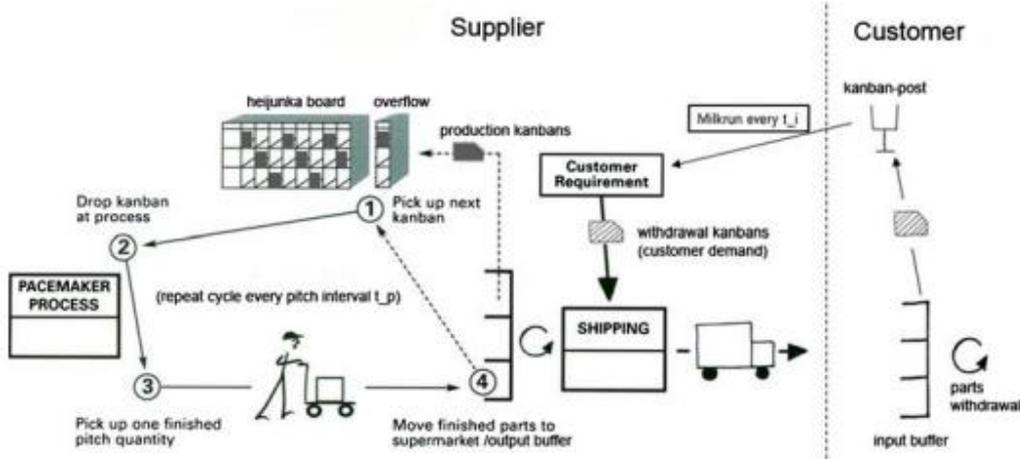


Ilustración 13: Funcionamiento del tablero heijunka. Fuente: (Matzka, Di Mascolo, & Furmans, 2012)

A partir de esta ilustración nos podemos hacer una buena idea de cómo funciona esta supuesta fábrica. En la planta de consumidores, las partes necesarias para producción son retiradas del buffer de entrada, por lo que los Kanban de transporte son despegados de estas partes y colocados en el buzón Kanban.

A continuación, el cliente pide partes a su proveedor en intervalos de tiempo  $t_i$  de acuerdo con el número de Kanban de transporte, que son cogidas de un inventario de bienes terminados siempre que estén disponibles.

Se utiliza el concepto de milk run, muy común en logística, que consiste en una técnica de transporte y entrega de materiales en la que un vehículo recoge múltiples suministros de diferentes proveedores en un solo viaje y los entrega en un destino final, generalmente una planta de producción o una tienda minorista. El objetivo principal del milk run es optimizar el uso de los recursos de transporte y reducir los costos asociados con el transporte y el almacenamiento de materiales. En lugar de enviar varios camiones de diferentes proveedores a una misma ubicación, se utiliza un solo vehículo para recoger todo lo necesario en un solo viaje.

Además de reducir los costos, el milk run también puede mejorar la eficiencia de la cadena de suministro al permitir una programación más precisa y reducir los tiempos de espera. Al recoger los materiales en un horario específico y programado, los proveedores pueden asegurarse de que los productos estén disponibles cuando se necesiten en la línea de producción o en la tienda minorista. Su funcionamiento se puede visualizar mejor con la imagen de (Brar & Saini, 2011):

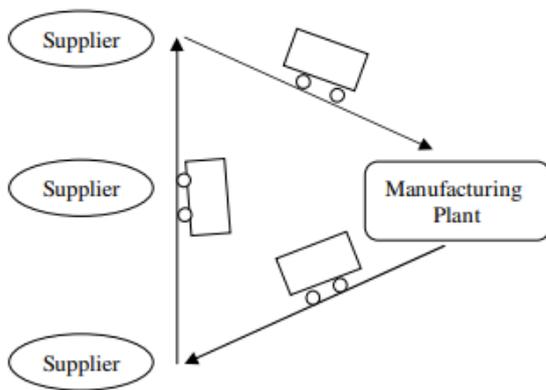


Ilustración 14: Funcionamiento milk run. Fuente: (Brar & Saini, 2011)

La demanda que no se ha conseguido cubrir tiene que esperar y las partes terminadas pueden ser enviadas al consumidor a través del próximo milkrun. Por otro lado, según los ítems que han sido retirados, los Kanban de producción son colocados en el tablero Heijunka.

Este tablero tiene una columna de Kanbans por cada intervalo de lanzamiento (“pitch Interval “en inglés), llamado  $tp$  y una fila por cada tipo de producto.

El tiempo de lanzamiento se refiere a la cantidad de tiempo necesaria para que un contenedor se llene de bienes terminados. Por ejemplo, si tenemos un tiempo de ciclo de 20 segundos, y el tamaño del contenedor es de 30 piezas, el intervalo es de 10 minutos. Entonces los Kanban de producción se depositan en el tablero heijunka en la secuencia deseada, según el mix de producción.

En caso de que lleguen un mayor número de tarjetas de la capacidad máxima del tablero Kanban, se sitúan en una localización determinada para ese exceso de número. Estas tarjetas se van moviendo al tablero heijunka a medida que éste empiece a liberarse paulatinamente.

Los espacios que no pueden ser rellenos dentro del tablero, se mantienen vacíos, para así no producir bienes que no son necesarios. El responsable de material coge un Kanban de producción del tablero heijunka y lo deposita en el proceso de producción. Este proceso se denomina “pacemaker process”, que en español es algo así como un proceso que nos marca el ritmo de producción, con lo que todos los procesos aguas arriba pueden ser configurados. De esta manera se puede crear un flujo de producción de una manera predecible.

A continuación, el responsable debería coger las partes terminadas del ciclo de producción precedente y moverlo al búfer de salidas, que es el supermercado. Este ciclo se repite en cada intervalo y se denomina  $tp$ .

## MODELO DE COLAS PARA EL MODELO

En este artículo (Matzka, Di Mascolo, & Furmans, 2012) proponen un modelo de colas de más de una clase, en el que varios productos se producen en una misma línea de producción. Según los autores, el número de óptimo de Kanban de cada uno de estos productos es independiente de los otros productos.

En cambio, las demandas de producción de cada producto, la secuencia de producción y el tiempo de transición solo influyen en el tiempo de ciclo de la producción, y en ningún parámetro más. Por consiguiente, se puede analizar el sistema heijunka fijándonos solo en un tipo de producto para obtener el número óptimo de Kanban para cada producto de manera separada. El modelo propuesto por (Matzka, Di Mascolo, & Furmans, 2012) se resume en la siguiente ilustración:

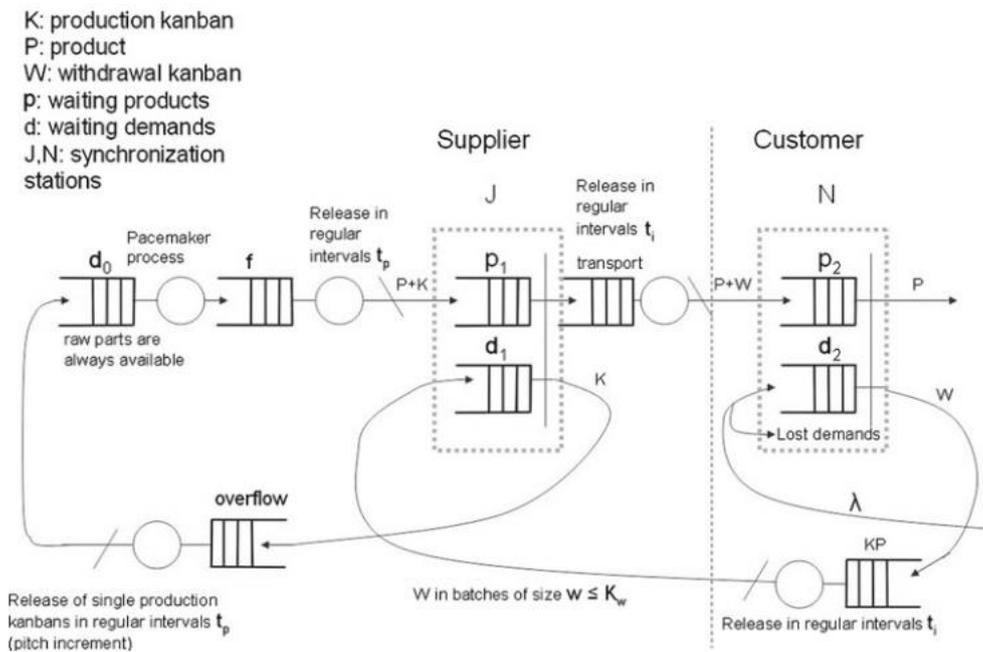


Ilustración 15: Modelo propuesto de un sistema heijunka Kanban. Fuente: (Matzka, Di Mascolo, & Furmans, 2012)

En esta figura se puede observar el funcionamiento del sistema propuesto, en el que solo se produce un único producto, para así poder nivelar el volumen de producción. En la estación  $N$  la demanda llega en la cola  $d_2$  en unidades sueltas, mediante una distribución de Poisson con una tasa  $\lambda$ .

Entonces, si la cola  $p_2$  está vacía, las demandas se pierden debido a que no se pueden completar de una manera inmediata, perdiendo capacidad de producción para el cliente, por lo que la cola  $d_2$  permanecerá siempre vacía.

En la cola  $p_2$  hay un número máximo de partes, denominado  $K_w$ , con los Kanban de transporte pegados, que pueden ser almacenadas. Entonces, si está disponible un bien terminado en la cola  $p_2$  cuando llega la demanda, la parte  $P$  es retirada, y el Kanban que tenía pegado se retira también y vuelve a ser almacenado en el buzón Kanban  $K_P$ .

Los Kanban de transporte son retirados del mismo supermercado en lotes de diferente tamaño  $w \leq K_w$  en intervalos de tiempo  $t_i$ . Esto sucede cuando se llevan hasta la planta de suministros y los productos terminados entran cada uno en su milkrun, asumiéndose que los tiempos de traslado son insignificantes.

Los lotes de los Kanban de transporte llegan a la cola  $d_1$ , cuya capacidad está limitada por un número  $K_p$  de partes  $P$ . Estas partes, con sus correspondientes Kanban de producción pueden ser almacenadas. Si los bienes terminados están disponibles en la cola  $p_1$ , la demanda se cumple y dichas partes pueden ser retiradas. Si hay las suficientes partes disponibles, la demanda pendiente tiene que esperar en la cola  $d_1$  hasta que lleguen nuevas piezas terminadas a la cola  $p_1$ .

A continuación, los Kanban de producción son despegados de las piezas terminadas que han sido retiradas, y esperan en la cola "overflow". En el caso de tener solo un producto, el tablero heijunka no es necesario, ya que ahora no tendríamos que ir anotando la secuencia de producción, ya que se trata de un único producto.

Para cada parte terminadas que es retirada, hay asociado un Kanban de transporte  $W$  desde la cola  $d_1$ , donde se pega a dicha parte. Luego, las partes que están esperando en la cola de transporte hasta que el responsable las lleva a lo largo de la milk run en intervalos de tiempo  $t_i$ .

Desde la cola de desbordamiento u "overflow", un Kanban de producción sale con un intervalo de tiempo  $t_p$ , y tiene que esperar en la cola  $d_0$  hasta que el servidor quede libre. En este problema (Matzka, Di Mascolo, & Furmans, 2012) asumen que siempre hay materia prima disponible, por lo que no hay ninguna estación de sincronización. Se da por hecho también que el proceso se lleva a cabo según el tiempo de ciclo, y el tiempo de procesado es constante.

Las partes terminadas esperan en la cola  $f$  hasta que el responsable de material las lleva al búfer de salida en intervalos de tiempo  $t_p$ . Se da por hecho que no hay tiempos ociosos ni ningún tipo de avería. Así que los Kanban retirados pueden ser procesados en un intervalo  $t_p$  y las partes terminadas están disponibles en el búfer de salida de la cola  $p_1$  al final del intervalo.

La cantidad de producción que se tiene que producir en cada intervalo se denomina  $H$ , y se obtiene a partir de redondear una demanda estimada ( $[E(w)]$ ) que puede llegar de cada milk run. Para simplificar el problema, se asume que los Kanban de producción no salen de la cola overflow uno por uno, sino que un número  $h \leq H$  de tarjetas Kanban de producción salen de esa cola en intervalos de tiempo  $t_i$ .

## NÚMERO ÓPTIMO DE KANBANS DE PRODUCCIÓN

El principal objetivo de este artículo es obtener el número óptimo de Kanban de producción  $K_p$  y el tamaño óptimo del búfer de salida, para poder garantizar un nivel de servicio lo suficientemente grande.

Primero tenemos que determinar los Kanban de transporte  $K_w$  en el bucle entre el cliente y el proveedor, y también la distribución de los lotes de las demandas entrantes, echando un vistazo a la estación de sincronización J. Es importante estar atento a su comportamiento ya que es el punto de encuentro entre los Kanban de producción y de transporte.

La distribución de los lotes de demanda entrante a la estación J se calcula mediante una función de poisson en la que tenemos una probabilidad de llegada dependiente de varios factores como los Kanban de transporte o el tiempo entre dos milk runs.

“El nivel de stock base del búfer de entrada de los clientes puede ser determinado con el número más pequeño de Kanban de transporte” garantizando un nivel de servicio del 99,9 %, por lo que así se puede obtener el número máximo de Kanban de transporte que pueden llegar a la estación J en cada milk run.

## NIVEL DE SERVICIO

Para la obtención de la fórmula del nivel de servicio es necesario fijar las probabilidades para cada estado  $x_j$ , que están definidas por la siguiente fórmula:

- $x_j = p(0, K_p - j)$  para  $j=0$  hasta  $K_p$
- $x_j = p(j, -K_p, 0)$  para  $j=K_p + 1$  hasta  $m$ , siendo  $m$  el número de filas de la matriz de transición.

Una vez que todas las demandas se han cumplido, sumando todas las probabilidades  $x_i$  para todos los estados  $i$ , entre 0 y  $K_p$ , obtenemos el nivel de servicio en el búfer de salida:

Se calcula como el sumatorio de todos los estados  $x_i$  desde  $i=0$  hasta  $K_p$

Una vez se realizan los cálculos para distintos números de Kanban de producción, encontramos un tamaño óptimo del búfer de salida,  $K_p \text{ opt}$ , que significa el  $K_p$  más pequeño que garantiza un nivel de servicio  $\alpha$ :

El algoritmo completo ha sido implementado con Matlab, obteniendo una serie de resultados numéricos. Además, para el caso de más de un producto se ha conseguido obtener el número de partes  $H_i$  de cada producto  $i$ , que debería ser producida en el intervalo entre dos milkruns.

También se obtendrá la cantidad óptima para cada producto, la configuración del tablero heijunka, y también el EPEI (el intervalo de tiempo en el que cada producto puede ser producido al menos una vez).

En el caso de que el intervalo milkrun sea un múltiplo del EPEI, no influye en el cálculo del número de kanbans necesarios, pudiéndose hacer por productos separados utilizando el Hi. En caso contrario, se deberán hacer modificaciones para poder calcular de manera acertada su número.

## RESULTADOS NUMÉRICOS

En este último apartado de artículo, (Matzka, Di Mascolo, & Furmans, 2012) tratan de comparar los resultados del método con resultados de simulación, para así verificar si se trata de un método acertado. También se comprobará cambiar distintos parámetros para así evaluar la precisión del método a la hora de calcular el número óptimo de kanbans.

Se toma para la distribución de poisson una lambda =1, y los clientes demandan tener en su búfer de entrada un nivel de servicio del 99,9 % (SLC). Además, los proveedores piden también tener un nivel de servicio del 95% en su búfer de salida (SLS).

SLS: Nivel de servicio en el output de proveedores (“Service Level at the suppliers output” en inglés)

SLC: Nivel de servicio en el input de los clientes.

Una vez programado el método en Matlab, se obtiene un Kanban de producción óptimo ( $K_p$  óptimo) de 72 y un número de kanbans de transporte ( $K_w$ ) en el bucle entre el consumidor y el proveedor de 5. El valor de H, que era la cantidad de producción planificada es  $H=1$  partes por cada intervalo milkrun  $t_i=1$ .

Estos parámetros se comparan con el lenguaje de simulación SIMAN y (Matzka, Di Mascolo, & Furmans, 2012) obtienen muy buenos resultados, con errores relativos muy pequeños como se muestran en la tabla:

$K_p$	$K_w$	SLS simu	Relative error in %	SLC simu	Relative error in %
73	5	0.95051		0.99322	
72	5	0.94980	-0,0211	0.99313	-0,5911
71	5	0.94910		0.99305	
72	4	0.97147		0.99113	
72	5	0.94980	-0,0211	0.99313	-0,5911
72	6	0.93534		0.99349	
71	4	0.97093		0.99105	
72	5	0.94980	-0,0211	0.99313	-0,5911
73	6	0.93624		0.99358	

*Tabla 6: Resultados comparativos con SIMAN. Fuente: (Matzka, Di Mascolo, & Furmans, 2012)*

Se lleva a cabo otro experimento para comprobar de otro modo la validez del método, por lo que se cambia el número de Kanban de producción y de transporte, para un  $t_i=1$ . Los resultados aportados por (Matzka, Di Mascolo, & Furmans, 2012) vienen resumidos en la tabla que viene a continuación:

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

$t_i = 1$	Desired SL		SLS = 95%		SLC = 95%	
	$K_p$	$K_w$	SLS simu	Rel. error (%)	SLC simu	Rel. error (%)
opt analy	20	3	0.9482	-0.1898	0.9662	0.017053
opt simu	21	3	0.9520		0.9670	
$t_i = 1$	Desired SL		SLS = 95%		SLC = 99.9%	
	$K_p$	$K_w$	SLS simu	Rel. error (%)	SLC simu	Rel. error (%)
opt analy	72	5	0.9498	-0.0211	0.9931	-0.5941
	73	5	0.9505		0.9932	

Tabla 7: Resultados cambiando número de Kanban. Fuente: (Matzka, Di Mascolo, & Furmans, 2012)

Podemos observar que manteniendo fijo el número de Kanban de transporte, el nivel de servicio en el output de los proveedores (SLS) aumenta cuando cogemos un número mayor de Kanban de producción. Sucede lo mismo con el nivel de servicio en la entrada de clientes (SLC), ya que un mayor número de demandas pueden llevarse a cabo dentro de cada milkrun, llenándose el búfer SLC.

Cunado reducimos el número de Kanban de transporte, como es lógico y observando los resultados de la tabla 1, el nivel de servicio para los clientes disminuye, ya que hay demandas que no pueden cumplirse de una manera inmediata.

Si reducimos ambos Kanban, en una unidad, el nivel de servicio en el búfer de salida de proveedores (SLS) todavía sigue aumentando, debido a que la influencia de una menor demanda entrante en cada intervalo milkrun es mayor que el efecto de un búfer de menor tamaño  $K_p$ .

A pesar de todo, los valores que se han calculado en la tabla 1 para el número de tarjetas Kanban de producción y de transporte no son los óptimos, aunque sí que se aproximan al nivel de servicio requerido.

Esto es debido, según (Matzka, Di Mascolo, & Furmans, 2012) a que, en el cálculo del número de lotes de transporte, no tenemos en cuenta que el búfer de entrada a clientes no permanece lleno después de cada milkrun, ya que precisamente no toda la demanda puede ser cumplida siempre.

Por lo tanto, no nos va a ser posible obtener el nivel de servicio que en un principio estaba estipulado con ese número de tarjetas Kanban de transporte. En la tabla realizada por (Matzka, Di Mascolo, & Furmans, 2012) que viene a continuación se expone el número de tarjetas Kanban óptimo para cada nivel de servicio SLS y SLC, siendo el siguiente:

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

$K_w$	2	3	3	3	3	3	3	3	4	5
SL customer	91%	92%	93%	94%	95%	96%	97%	98%	99%	99.9%
SL supplier	$K_p$									
SLS 90%	4	12	12	12	12	12	12	12	24	35
SLS 91%	5	14	14	14	14	14	14	14	27	39
SLS 92%	5	15	15	15	15	15	15	15	30	45
SLS 93%	5	16	16	16	16	16	16	16	34	51
SLS 94%	6	18	18	18	18	18	18	18	39	60
SLS 95%	6	20	20	20	20	20	20	20	46	72
SLS 96%	7	23	23	23	23	23	23	23	55	90
SLS 97%	7	28	28	28	28	28	28	28	69	119
SLS 98%	9	34	34	34	34	34	34	34	93	173
SLS 99%	11	46	46	46	46	46	46	46	144	315

*Tabla 8: Número de Kanban para cada nivel de servicio. Fuente: (Matzka, Di Mascolo, & Furmans, 2012)*

Los valores obtenidos en esta tabla son ligeramente mejores respecto al método propuesto, pero el tiempo de cálculo es mucho mayor que utilizando el método matemático, ya que se trata de una simulación.

## CONCLUSIONES

En este artículo se ha calculado el número de Kanban en un sistema heijunka Kanban obteniendo buenos resultados mediante un modelo matemático que se ha ido explicando paso a paso a lo largo del mismo. Una vez implementado el problema, se ha obtenido un número de Kanban de transporte y de Kanban de producción óptimo para un nivel de servicio del cliente y del proveedor determinado.

Por último, se ha experimentado con el número de Kanban de transporte que serían necesarios para satisfacer el nivel de servicio del proveedor desde valores del 90% al 99%.

En resumidas cuentas, cabe destacar que este método se puede utilizar en una planta multiproducto, calculando el número de Kanban por separado para cada producto, pero única y exclusivamente si el intervalo milkrun es múltiplo del valor calculado EPEI, que era el tiempo necesario para fabricar un producto al menos una vez.

### 3.2.2 PROGRAMACIÓN DINÁMICA PARA UNA PRODUCCIÓN DE LÍNEA DE UN PRODUCTO CON VARIAS ETAPAS CONTROLADA POR KANBAN

En este artículo de (Al-Tahat, Dalalah, & Barghash, 2012) se va a describir una producción de línea de un producto con varias etapas en la que el objetivo principal es determinar una serie de procedimientos para poder ir cambiando el número y el tamaño de los Kanban, con el objetivo de llevar a cabo una programación dinámica para una determinada línea de producción

#### NOMENCLATURA

A lo largo de este artículo se utilizan los siguientes términos:

N: Número total de estaciones de trabajo en la línea de producción

KPi: Número de kanbans de producción en la estación  $i$

Kwi: Número de kanbans de transporte en la estación  $i$

Ki: Número de kanbans en la estación  $i$

ni: Tamaño de Kanban en la estación  $i$

Dc: Tasa de demanda del cliente

Di: Demanda en la estación de trabajo  $i$

SSi: Stock de seguridad en la estación de trabajo  $i$

$\mu_i$ : Tasas de procesamiento totales en la etapa  $i$

Kc: Número de Kanban de transporte del cliente

nk: tamaño del Kanban de transporte del cliente

$\lambda_i$ : tasa de llegada del trabajo en curso (WIP) en la estación de trabajo  $i$

$\rho_i, \rho(i)$ : Porcentajes de uso de la estación de trabajo  $i$

Si, S(i): Número de procesadores instalados en la estación de trabajo  $i$

$\mu_{si}$ : Tasas de procesamiento para cada procesador en la estación de trabajo  $i$

WIPi: Número total de unidades de trabajo en curso (WIP) integradas hasta la estación de trabajo  $i$

Lsi: Nivel de trabajo en curso esperado en cada momento (unidades)

$\delta_i$ : Niveles de trabajo en curso en la estación de trabajo  $i$

$\rho_{\delta i}$ : Probabilidades de tener  $\delta$  unidades de trabajo en curso (WIP) en la estación de trabajo  $i$ .

$p_{0i}$ : Probabilidades de tener 0 unidades de trabajo en curso (WIP) en la estación de trabajo  $i$ .

$L_{si}$ : Tiempo de espera esperado del trabajo en curso (WIP) en cada momento (unidades de tiempo)

$t_i$ : Tiempo de espera hasta la cola  $q_i$  en la estación de trabajo  $i$

$b_{oi}$ : Coste de comprar una unidad de materia prima en la estación de trabajo  $i$

$D_{oi}$ : Unidades compradas de materia prima en la estación de trabajo  $i$

$C_{1i}$ : Coste fijo por cada viaje para mover el trabajo en curso (WIP) en la estación de trabajo  $i$

$C_{2i}$ : Coste variable por cada viaje para mover el trabajo en curso (WIP) en la estación de trabajo  $i$

$p_i$ : Coste de procesado de una unidad en la estación  $i$

$f_i$ : Coste fijo de instalación para un servidor en la estación de trabajo  $i$

$h_i$ : Coste de inventario en la estación de trabajo  $i$

$TC_i$ : Coste operacional total acumulado del sistema hasta la estación de trabajo  $i$

## MODELO DE PROGRAMACIÓN DINÁMICO

En numerosos artículos aparte de este mismo, se determina el número y tamaño de las tarjetas Kanban, asumiendo que nunca hay escasez en la demanda. Esta afirmación implica que el inventario objetivo en cualquier etapa debería ser igual al tiempo de espera de la demanda, para así conseguir cierta seguridad en el stock del inventario.

Esto da lugar a la siguiente fórmula del tiempo de espera:

$$T_i = \frac{k_i n_i}{D_{i+1} + ssi}$$

- $k_i$ : Número de Kanban en la estación  $i$
- $n_i$ : Tamaño de Kanban en la estación  $i$
- $D_{i+1}$ : Demanda en la siguiente estación de trabajo  $i+1$
- $ssi$ : stock de seguridad en la estación  $i$

Una vez tenemos la tasa de demanda del cliente, el número de Kanban y su tamaño se puede expresar de la siguiente manera:

$$kc \cdot nc = Dc$$

- kc: Número de Kanban de transporte del cliente
- nc: tamaño de Kanban de transporte del cliente
- Dc: Tasa de demanda del cliente

A continuación, se calculan las unidades de trabajo en curso y el tiempo de espera de cada Kanban. Las unidades de trabajo en curso están determinadas por factores como los niveles de trabajo en curso esperados en cada momento. Para los tiempos de espera influye  $\lambda_i$ , que es la tasa de llegada del trabajo en curso (WIP) en la estación de trabajo  $i$

Para la resolución del problema por programación dinámica, lo primero que hacen (Al-Tahat, Dalalah, & Barghash, 2012) es descomponer la línea de producción en distintas etapas. A continuación, se lleva a cabo el problema matemático de programación directa.

Estableceremos que  $WIP_i$  se refiere al número total de unidades WIP en el sistema productivo hasta la cola  $i$  de la estación de trabajo  $i$ . Se definirá  $L_{si}$  como el número de unidades de trabajo en curso en la cola  $i$ . En este problema se computará  $WIP_i$  a partir de  $WIP_{i+1}$ , donde  $i$  va desde 1 hasta el número total de estaciones de trabajo  $N$ , y el trabajo en curso para la estación  $N+1$  es igual a cero, ya que, al finalizar todas las estaciones de trabajo, ya no hay trabajo en curso pendiente.

Cabe destacar que en este artículo se utiliza el modelo de colas M/M/s. Este modelo matemático se utiliza para la teoría de colas, en la que la tasa de servicio y la tasa de llegadas siguen una distribución exponencial y tenemos un total de  $s$  servidores.  $M$  nos indica que las llegadas siguen una distribución de Poisson, y que el tiempo de servicio sigue una distribución exponencial.

El modelo asume entonces, que la capacidad de la cola es ilimitada, por lo que no hay límite en el número de clientes que tienen que esperar.

Tradicionalmente, el tamaño de Kanban se ha fijado de una manera conveniente a la cantidad que mejor se adapta al tamaño de los contenedores de materiales. El nivel de inventario medio del búfer se calcula a partir del nivel máximo de inventario menos el número de unidades en orden.

Su nivel máximo se determina con la fórmula  $K_i \cdot n_i$ , donde  $K_i$  es el número de Kanban en la etapa  $i$ . Las unidades en orden de las que acabo de hablar vienen determinadas por la siguiente fórmula:

$$\frac{n_i - 1}{2} \text{ unidades en la etapa } i$$

Para concluir con las operaciones matemáticas, el sistema desarrollado hasta el momento nos proporciona el coste operacional total, el cual se va a evaluar en los resultados numéricos del modelo.

## RESULTADOS NUMÉRICOS

El sistema desarrollado consta de 3 etapas. Dicho sistema trabaja bajo las siguientes condiciones, y se adjunta una imagen aportada por (Al-Tahat, Dalalah, & Barghash, 2012), en la que se muestran los resultados del TC óptimo.

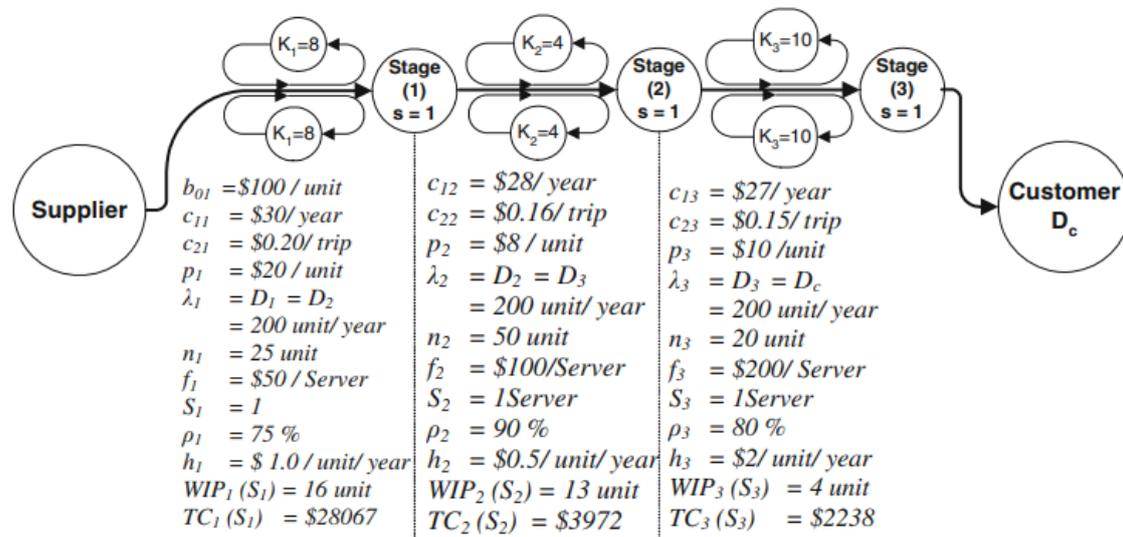


Ilustración 16: Resumen de resultados. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012)

A partir de estos resultados, se analizan una serie de efectos que puede tener sobre el sistema el cambio en el valor de alguno de los parámetros.

Un aumento en el número de servidores hace reducir los tiempos de espera y la cantidad de trabajo en curso en la estación de trabajo  $i$ , ya que se fomenta la instantaneidad. Por otro lado, y como cabía esperar, aumenta el coste operacional total (TC3, ya que se trata de la etapa número 3), como se puede ver en la siguiente tabla realizada por (Al-Tahat, Dalalah, & Barghash, 2012):

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

Stage 3: $TC_3(S_3)$																		
$D_{3+1}$	$D_3 = \lambda_3$	$\rho_3$	$\mu_3$	$n_3$	$K_3$	$S_3$	$sub$	$P_{03}$	$WIP_3$	$\tau_3$	$c_{13}$	$c_{23}$	$D_{o3}$	$b_{o3}$	$p_3$	$f_3$	$h_3$	$TC_3(S_3)$
200	200	0.8	250	20	10	1	1.0	0.20	4	0.0200	27	0.15	0	0	10	200	2	2238
						2	0.8	0.43	2	0.0115								2435
						3	0.3	0.45	2	0.0076								2633
						4	0.1	0.45	1	0.0052								2832
						5	0.0	0.45	1	0.0043								3032

*Tabla 9: Efecto en el aumento de número de servidores para la etapa número 3.  
Fuente: (Al-Tahat, Dalalah, & Barghash, 2012)*

La elección del tamaño del contenedor influye también en gran medida en el número de Kanban necesarios. Cuanto más grande sea el contenedor, necesitaremos un menor número de Kanban. Los otros parámetros estudiados, como el tiempo de espera, el trabajo en curso y el coste operacional total permanecen prácticamente sin variaciones, como se aprecia en la tabla de (Al-Tahat, Dalalah, & Barghash, 2012):

Stage 3: $TC_3(n_3)$																		
$D_{3+1}$	$D_3 = \lambda_3$	$\rho_3$	$\mu_3$	$n_3$	$K_3$	$S_3$	$sub$	$P_{03}$	$WIP_3$	$\tau_3$	$c_{13}$	$c_{23}$	$D_{o3}$	$b_{o3}$	$p_3$	$f_3$	$h_3$	$TC_3(n_3)$
200	200	0.8	250	15	13	1	1	0.20	4	0.0200	27	0.15	0	0	10	200	2	2239
				25	8			0.20	4	0.0200								2237
				35	6			0.20	4	0.0200								2237
				45	4			0.20	4	0.0200								2236
				50	4			0.20	4	0.0200								2236

*Tabla 10: Efecto del tamaño del contenedor en los resultados de la etapa 3.  
Fuente: (Al-Tahat, Dalalah, & Barghash, 2012)*

Hay muchos cambios de parámetros a estudiar, pero para concluir se va a exponer el efecto que supone un cambio en la tasa de demanda. A mayor tasa de demanda, necesitaremos un mayor número de tarjetas Kanban, produciéndose un aumento en el coste operacional total.

#### VALIDACIÓN DE RESULTADOS MEDIANTE SIMULACIÓN

Para la comprobación de resultados (Al-Tahat, Dalalah, & Barghash, 2012) utilizaron la herramienta de simulación ProModel, donde los procesadores se programan como localizaciones y las tarjetas de transporte, órdenes y tarjetas de producción se modelan como entidades.

En este caso el valor del coste variable  $TC_i$  se fija en un valor inicial  $c_{2i} + f_i$ , y va siendo modificado siguiendo una trayectoria y llegada Kanban en una etapa con un valor de  $c_{2i}$ .

Una variable denominada partes en la célula (“parts\_in\_cell “en inglés) es definida y aumenta su valor en una unidad en cuanto una entidad llega a las inmediaciones de la célula  $i$ , y su número se ve reducido en una unidad en cuanto una parte deja las inmediaciones de la célula. (Al-Tahat, Dalalah, & Barghash, 2012) han incluido una captura de pantalla del modelado, que es la imagen siguiente:

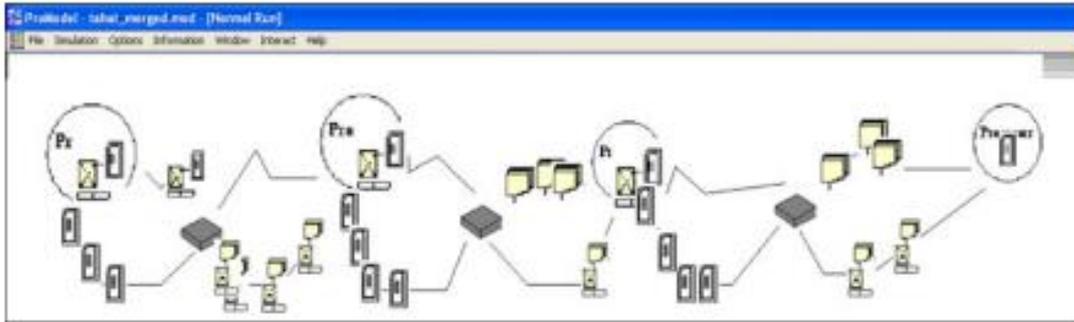


Ilustración 17: Modelo utilizado. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012)

A continuación, se exponen una serie de tablas al igual que con el modelo matemático, donde se puede comprobar que hay errores muy pequeños entre el modelo de simulación y el modelo matemático, siguiendo los mismos patrones a la hora de modificar parámetros.

Para la etapa 3 se puede comprobar a partir de la siguiente tabla aportada por (Al-Tahat, Dalalah, & Barghash, 2012):

Stage 3: $TC_3(S_3)$																		
$D_{3+1}$	$D_3 = \lambda_3$	$\rho_3$	$\mu_3$	$n_3$	$K_3$	$S_3$	sub	$P_{03}$	WIP <sub>3</sub>	$\tau_3$	$c_{13}$	$c_{23}$	$D_{03}$	$b_{03}$	$p_3$	$f_3$	$h_3$	$TC_3(S_3)$
200	200	0.8	250	20	10	1	1.0	0.20	5.51	0.0200	27	0.15	0	0	10	200	2	2234
						2	0.8	0.43	1.15	0.0115								2428
						3	0.3	0.45	0.88	0.0076								2626
						4	0.1	0.45	0.78	0.0052								2826
						5	0.0	0.45	0.78	0.0043								3026

Tabla 11: Resultados para el cambio de número de servidores, en esta ocasión para el modelo de simulación. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012)

Si lo comparamos con los resultados obtenidos con el método matemático, como ya hemos apuntado antes, las diferencias son casi indescriptibles, por lo que eso nos puede dar una idea de la validez del modelo.

## CONCLUSIONES

Se ha utilizado un método matemático para determinar el número de Kanban, y un modelo de simulación para comprobar la calidad de las soluciones. Se ha utilizado un modelo de colas dinámico M/M/s, y se ha calculado el coste operacional total,  $TC_i$ , para cada una de las tres etapas de las que consta el problema.

Además, se ha ido haciendo una comparación de los resultados al alterar el valor de algunos parámetros, como por ejemplo el tamaño del contenedor o el número de servidores disponibles. En definitiva, este modelo resulta ser muy fiable, ya que los resultados arrojan un error relativo bastante pequeño (4,4%).

## 3.3 MÉTODOS DE SIMULACIÓN

Un modelo de simulación es una representación, ya sea matemática, física o informática de un sistema o un proceso real que se utiliza para analizar su comportamiento y predecir su desempeño en diferentes escenarios o condiciones.

El modelo simula el comportamiento del sistema o proceso utilizando diferentes variables y parámetros, y se pueden realizar diferentes pruebas y análisis para evaluar cómo el sistema o proceso podría responder a diferentes situaciones.

Hay muchos programas que se encargan de simular estos modelos, que se pueden utilizar en numerosos campos y disciplinas aparte del objetivo principal de este trabajo, que es determinar el número de tarjetas Kanban. Entre estos programas están el propio Microsoft Excel, Arena, eKanban, Trello, etc.

Este apartado va a tratar sobre un artículo de un estudio de simulación, que es distinto a los artículos anteriores, ya que incluye al método EPQ (Economic Production Quantity), de (Moran & Brayer, 2011), un método de cálculo de tamaño de lote.

### 3.3.1 ESTUDIO DE SIMULACIÓN EPQ Y KANBAN

En el artículo de (Moran & Brayer, 2011) se realiza un estudio sobre un sistema de simulación con el objetivo principal de reducir el coste anual de puesta en marcha (en inglés “setup”), que se puede aproximar como todos los costes incurridos para la configuración la maquinaria involucrada en la producción.

Se estudian los modelos EPQ y Kanban, siendo el EPQ la variante del más conocido EOQ (Economic order Quantity), cuyo cálculo se lleva a cabo con la fórmula de Wilson.

Fórmula EOQ:

$$Q = \sqrt{\frac{2 * ce * D}{cp * \theta}}$$

, donde ce es el coste de emisión, D la demanda total, cp el coste de posesión y  $\theta$  el número de semanas.

La gran diferencia entre EPQ y EOQ es que en el modelo EOQ las órdenes son todas recibidas a la vez, mientras que en el EPQ son recibidas de una forma paulatina a lo largo del tiempo.

El EPQ (Economic Production Quantity) es un modelo matemático utilizado en la optimización de inventarios utilizado para determinar el tamaño óptimo de lote de producción, para así disminuir los costes de almacenamiento y producción.

Este modelo solo tiene en cuenta el coste de producción de los productos, el coste de almacenamiento de los inventarios, y la tasa de demanda de los productos para determinar la cantidad óptima de producción. Se supone siempre que hay una demanda constante y que el tiempo de producción es conocido.

El objetivo principal de este artículo es evaluar EPQ y Kanban en una línea de producción, bajo un sistema en el que se producen varios productos con varios tiempos de puesta en marcha (“setup times” en inglés), que es la variable principal. La medida de rendimiento es la que se ha comentado antes: el coste anual de puesta en marcha. Esto permitirá averiguar qué sistema de los dos es más conveniente según las circunstancias del mercado para conseguir una mayor competitividad.

Se utilizará el sistema Arena para la simulación, cuyos resultados serán analizados estadísticamente también con Arena. La medida de rendimiento es el “setup cost”, que (Moran & Brayer, 2011) lo definen como los costes incurridos mientras se prepara el sistema de producción para otro producto.

Durante el ejercicio de un año, esto da lugar al “annual setup cost”, que se define como el número total de setups para producir la demanda anual de todos los productos de la línea de producción, siendo de 60\$ la hora.

## METODOLOGÍA

Como se ha apuntado anteriormente, se utiliza como variable el setup time, y el lenguaje de simulación es Arena. Las combinaciones que se estudian por (Al-Tahat, Dalalah, & Barghash, 2012) para los EPQ y los Kanban son las siguientes:

System	Variable	Performance Metric
1. EPQ	1. Setup Time: 1, 5, 10, 15, 30, 60 Minutes	1. Annual setup cost
2. Kanban	2. Setup Time: 1, 5, 10, 15, 30, 60 Minutes	2. Annual setup cost

Tabla 12: Variables utilizadas en la simulación. Fuente: (Al-Tahat, Dalalah, & Barghash, 2012)

Tanto Kanban como EPQ van a servir para emular la fabricación de un producto terminado, que no necesita ensamblaje. Se va a utilizar un tamaño de contenedor de 5,10 y 15 unidades respectivamente, para así investigar las fluctuaciones que pueda haber al cambiar el tamaño de estos. La línea de producción a simular consta de 5 estaciones de trabajo, suficiente para representar un supuesto real.

En el caso de Kanban, se utiliza un sistema de doble tarjeta, es decir, teniendo dos tipos: Kanban de producción y Kanban de transporte, habiendo dos buffers entre cada dos estaciones de trabajo.

En EPQ, podemos ver la representación del inventario de trabajo en curso, especificándose el número de unidades que tienen que ser completadas entre cada uno de los setups. Una vez se ha finalizado el lote al completo, la estación de trabajo experimentará un setup para el producto siguiente en la línea de producción, como se aprecia en la imagen de (Moran & Brayer, 2011):

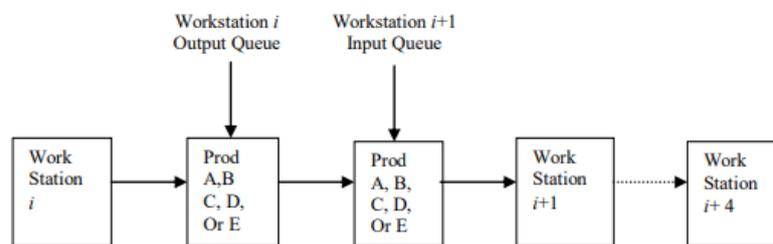


Ilustración 18: Representación EPQ. Fuente: (Moran & Brayer, 2011)

## TÉCNICA ESTADÍSTICA

En este artículo se va a utilizar en análisis de varianza (ANOVA), ya que el objetivo principal no es determinar la naturaleza independiente de cada uno de los modelos, si no determinar si hay o no hay relación existente. Las hipótesis del modelo se utilizan contrastando individualmente los efectos de los setup time tanto de Kanban como de EPQ.

El nivel de significación utilizado para pruebas estadísticas de factor principal es un  $\alpha$  de 0,05. Como siempre en estos casos, hay dos alternativas para la prueba estadística:

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

- Hipótesis nula (Ho): Efecto del factor =0, el factor no tiene efecto
- Hipótesis alternativa (Ha): Efecto del factor  $\neq 0$ , el factor sí que tiene efecto.

La regla de decisión es la siguiente:

- Si el p-valor  $\geq 0,05$ , concluimos Ho
- Si el p-valor  $< 0,05$ , concluimos Ha

El número de réplicas para obtener la precisión deseada es calculado por el software Arena, con una confianza del 95%.

### PARÁMETROS UTILIZADOS

Los parámetros utilizados por (Moran & Brayer, 2011) para ambos modelos son los siguientes:

Parameter	Values
Demand	2000 units a year for each product 1 year = 50 weeks; 1 week = 5 days; 1 day = 24 hours
Setup Costs	\$60.00 per hour
Annual Holding Cost Rate	20%
Production Cost	\$80.00 per unit Each of the 5 workstations adds \$16.00 of costs
Processing Time	20 minutes per part total 4 minutes per workstation, gamma distribution $\alpha = 2, \sigma^2 = 8$
Number of Work Stations	5
Number of Products	5

Tabla 13: Parámetros aplicables para EPQ y Kanban. Fuente: (Moran & Brayer, 2011)

Parameter	Values
Setup Time (Primary variable)	1, 5, 10, 15, 30, 60 minutes. With a gamma distribution, $\alpha = 2$ .
Process Batch Size	EPQ: 84, 188, 266, 325, 460, 651 (based on EPQ formula, refer to Nicholas, 1998). Batch sizes correspond to setup times of 1, 5, 10, 30, and 60 minutes respectively.
EPQ (Transfer Batch)	1 unit

Tabla 14: Variables utilizadas en EPQ. Fuente: (Moran & Brayer, 2011)

Parameter	Values
Setup Time (Primary variable)	1, 5, 10, 15, 30, 60 minutes. With a gamma distribution, $\alpha = 2$ .
Kanban Container Size	5, 10, 15 units
Process Batch Size	Number of production kanban: 1 kanban per work station Number of withdrawal kanban: 1 kanban per work station (based on kanban formula, refer to Nicholas, 1998).
Kanban (transfer batch), which is kanban container size	5, 10, and 15 units per kanban container

Tabla 15: Variables utilizadas en Kanban. Fuente: (Moran & Brayer, 2011)

## COMPARACIÓN EPQ Y KANBAN

Tanto EPQ como Kanban son dos sistemas completamente distintos y las comparaciones directas pueden tener gran dificultad. En cambio, se puede hacer una comparación general entre ambos sistemas.

La primera pregunta que nos debemos hacer es que si el setup time afecta al coste anual de puesta en marcha (anual setup cost) en Kanban y en EPQ. Para ello comparamos en una gráfica el sistema EPQ y el sistema Kanban con tres variantes distintas: tamaños de contenedor de 5,10 y 15 unidades.

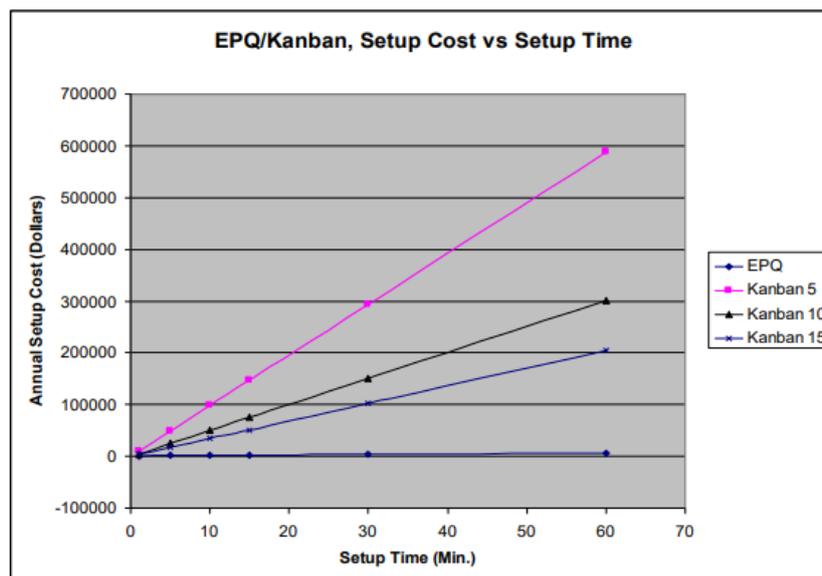


Ilustración 19: Resultados Annual setup cost vs setup time. Fuente: (Moran & Brayer, 2011)

Podemos comprobar que a medida que aumenta el setup time, aumenta nuestro coste de setup anual. El menor coste de EPQ respecto a cualquier tamaño de contenedor de Kanban se puede explicar ya que a medida que aumentamos el setup time, los tamaños de lote de EPQ se hacen mucho más grandes que los de Kanban.

Esto tiene sentido porque cuanto más grande sea el tamaño de lote habrá que hacer menos cambios en el setup, disminuyendo el coste anual del setup.

## CONCLUSIONES

En este artículo en el que se ha hecho un estudio comparando el método EPQ con el Sistema Kanban se ha llegado a la conclusión de que EPQ es más efectivo cuando lo comparamos con tamaños de contenedores Kanban pequeños, en el caso de la simulación de 5,10 y 15 unidades.

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

En cambio, para tamaños de contenedor más grandes es conveniente la elección del sistema Kanban debido a su gran flexibilidad.

## 4. MODELO DE SIMULACIÓN

### 4.1 INTRODUCCIÓN Y ELECCIÓN DE SIMULADOR

En este capítulo se va a ir explicando paso a paso todo lo relacionado con el modelo de simulación que se ha realizado, incluyendo un modelo preliminar y una mejor del modelo, con el objetivo de calcular el número óptimo de tarjetas Kanban en una línea de producción.

En este modelo no se distingue entre Kanban de producción o Kanban de transporte, a medida que avanza el proceso los lotes van pasando a lo largo de las zonas cada uno con una tarjeta Kanban asociada.

Como primer paso, se va a proceder a describir el modelo y todos los elementos de los que consta, detallando las macros de Excel utilizadas y todas las fórmulas necesarias para el buen funcionamiento del modelo. A continuación, expondré la simulación del modelo una vez se haya iniciado, para ver cómo podrían ir modificándose los parámetros y los efectos que tendría en los resultados que salen por pantalla.

Para finalizar, haré una discusión sobre los resultados obtenidos, mostrando distintos gráficos y tablas para una comprensión mucho más visual de todo el modelo en sí.

En este modelo se va a utilizar el programa Microsoft Excel para la simulación en este trabajo. Como todos sabemos, Microsoft Excel es un programa utilizado mayoritariamente para realizar hojas de cálculo, permitiendo un buen análisis de datos, la posibilidad de poder trabajar en tiempo real desde otros dispositivos y además incluye la opción de realizar gráficos de una manera muy sencilla.

El lenguaje utilizado en las macros de Excel es Visual Basic for Applications, que tiene su propio código, distinto a otros lenguajes de programación que de hecho se ha utilizado durante el estudio del grado, como pueden ser C++ o Python.

## 4.2 MODELO PRELIMINAR

El modelo que ha sido elaborado supone una línea de producción cualquiera con tres centros de trabajo. El primer centro de trabajo (CT1) se encargará de fabricar lotes de tipo “P”, y el segundo centro de trabajo (CT2) estará destinado a la elaboración de lotes de tipo “Q”. Ambos centros de trabajo surtirán de esos lotes al último centro de trabajo, CT3, en el que se ensamblarán las piezas y podrán ser enviadas al cliente.

En el modelo para cada lote distinto hay asignada una tarjeta Kanban, por lo que cada vez que veamos una letra en el modelo, ya sea P, Q o R dependiendo del centro de trabajo en el que nos encontremos, significará un lote en circulación al que se le asigna una tarjeta Kanban.

Como se ha apuntado anteriormente, el lenguaje utilizado es Visual Basic for Applications, necesario para conseguir la simulación del modelo. Su utilización es vital para el primer paso, donde seleccionamos la velocidad que queremos que tenga el modelo. En un principio, un segundo de la realidad supone un minuto en la simulación.

Para el desarrollo de las macros es clave activar la ventana de programador. Esto se hace en un simple paso dentro de la hoja de cálculo:

Archivo -> Opciones -> Personalizar cinta de opciones. Una vez que estamos en esta última ventana clicamos sobre el botón de la derecha que pone Programador. Con ello podremos añadir también los botones de los que vamos a hablar a continuación dentro de la pestaña Programador -> Insertar -> Botón de comando.

La configuración del modelo a primera vista es la siguiente:

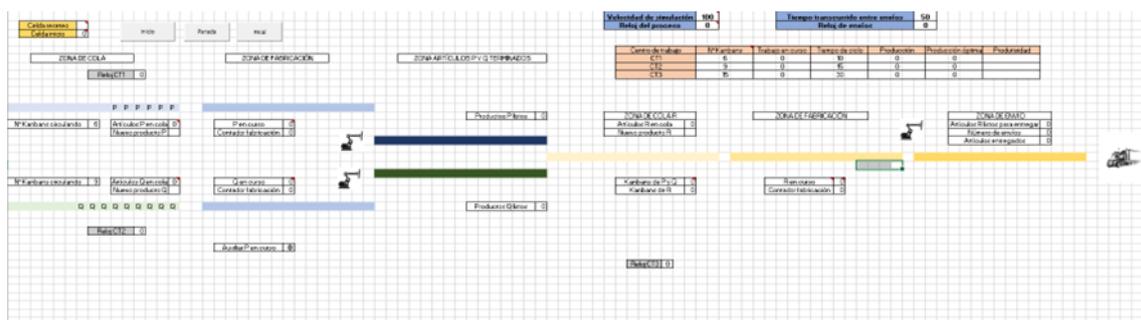


Ilustración 20. Modelo de simulación

Con una macro de Excel se ha conseguido que se pueda seleccionar la velocidad del modelo, es decir, poder determinar cuántos minutos pasan en la simulación cuando pasa solamente un segundo en la realidad.

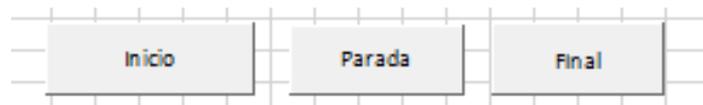
Por lo tanto, si en la casilla de velocidad de simulación ponemos el número 20, quiere decir que por cada segundo que pase en la realidad, pasan 20 minutos en la simulación, y viceversa. En mi caso es recomendable poner un número superior para así ver más rápido cómo va evolucionando la simulación del modelo, mostrándose dicha celda a continuación:

<b>Velocidad de simulación</b>	<b>100</b>
<b>Reloj del proceso</b>	<b>0</b>

*Tabla 16: Celdas donde se especifica la velocidad, y se muestra el reloj del proceso por pantalla.*

La casilla de abajo llamada “Reloj del proceso”, va acumulando el tiempo ficticio que pasa en la línea de producción hasta que se decida parar. Luego puede continuar, o puede reiniciarse. Una vez optamos por el reinicio, todos los valores que han ido saliendo por pantalla vuelven a valer 0, incluido el reloj.

Para que todo ello funcione, hay tres botones distintos. Cuando se utiliza la función de programador en Excel un botón es un objeto en el que en caso de que decidamos clicarlo, comienza un evento, o realiza cualquier cálculo, etc. En el caso del modelo de simulación, una vez se cliquen uno de esos tres botones pueden pasar tres eventos distintos: Inicio del proceso, parada del proceso y reinicio del proceso, que se representan de la siguiente manera:



*Ilustración 21: Botones del modelo de simulación.*

### Inicio del proceso

Una vez estamos en la pestaña de programador, para poder insertar un botón clicamos en insertar y dentro de las opciones la de botón de comando. Cuando ya se ha creado, si clicamos dos veces, nos lleva a la ventana del código de VBA, y nos crea una función asociada al botón. Esto lo hace cada vez que creamos cada uno de los tres botones, y nos permite escribir el código que queremos que se ejecute una vez se pulse ese botón.

Para determinar que comienza la simulación es necesario pulsar sobre el botón inicio, que está en la parte superior izquierda del modelo. Este botón cambiará el valor de la celda G3, comenzando a valer 1. Esto significa que la macro de Excel

comienza a funcionar, ya que cuando es igual a 0 se detiene, y cuando es igual a 1 aumenta el tiempo real de un segundo al modelo. Esas celdas son las siguientes:

Celda reseteo	
Celda inicio	0

Tabla 17. Celdas G2 y G3.

El código para el botón de inicio que se ha programado es el siguiente:

```
Private Sub Inicio_Click()  
    Range("g2").Value = ""  
    Range("g3").Value = 1  
    Kanban  
End Sub
```

Ilustración 22: Código de botón de inicio.

Esta fórmula se asegura que G2 esté vacía, y asigna el valor 1 a la celda G3. Una vez que ocurre esto llama a la función Kanban, que es la primera macro de la que se ha hablado, y que permite que funcione el reloj.

#### Parada del proceso

Cuando este botón es pulsado, cambia el valor de la celda G3 a cero, lo que hace que se detenga la función Kanban. Esto se utiliza cuando quiere pararse la simulación para poder comprobar cualquier tipo de resultado, pero en cualquier caso no reinicia ningún valor, simplemente para el proceso hasta en caso de que se quiera que continúe se pulse el botón inicio, mostrando su código a continuación:

```
Private Sub Parada_Click()  
    Range("g3").Value = 0  
End Sub
```

Ilustración 23: Código botón de parada.

### Final del proceso

En cuanto se pulsa este botón se reinician todos los valores. Esto se debe a que todas las celdas de la hoja de cálculo funcionan solamente si el valor de la celda G2 está vacío. En cuanto cambiamos el valor de esa celda, el elemento Calculate recalcula todos los valores, y a continuación la celda G2 pasa a valer cero de nuevo.

```
Private Sub Final_Click()  
    Range("g2").Value = "r"  
    Calculate  
    Range("g2").Value = " "  
End Sub
```

*Ilustración 24: Código botón de reinicio.*

Aparte de estas macros, la macro que permite que el proceso se ejecute y vaya avanzando, es la siguiente:

```
Private Sub Kanban()  
    For a = 1 To Range("b11").Value  
        If Range("g3").Value = 0 Then End  
        Calculate  
    Next a  
    If Range("g3").Value = 1 Then  
        Application.OnTime Now + TimeValue("00:00:01"), "Kanban"  
    End If  
End Sub
```

*Ilustración 25: Código macro Kanban*

Esta macro tiene un bucle for utilizado para que cada segundo que pase en la realidad consiga sumar el tiempo de ejecución según el número que hayamos indicado. Esto lo realiza cada segundo para poder ir actualizando el valor del reloj, siempre y cuando el valor de G3 sea igual a 1, ya que si no entraría en un bucle infinito que por mucho que diéramos al botón de parada o de reinicio seguiría sumando tiempo al reloj.

Cuando el valor de la celda G3 es igual a cero, lo que equivale a que se ha pulsado o el botón de parada o el de reinicio, el bucle for concluye.

## NÚMERO DE TARJETAS KANBAN Y TIEMPO DE CICLO

En el modelo hay una tabla en la que se pueden modificar dichos parámetros, mostrándose a continuación.

Centro de trabajo	Nº Kanbans	Trabajo en curso	Tiempo de ciclo
CT1	6	0	10
CT2	9	0	15
CT3	15	0	30

*Tabla 18: Celdas donde se establece el número de tarjetas y el tiempo de ciclo.*

Se ha establecido que en cada centro de trabajo el número máximo de tarjetas Kanban que pueden asignarse es de 15, ya que en la representación del modelo no hay espacio para más tarjetas, hay un máximo de 15 posiciones.

El número mínimo que se puede fijar es de una tarjeta para el primer centro de trabajo y de dos tarjetas para el segundo de trabajo. Esto está provocado por la limitación en las fórmulas elaboradas, ya que no se puede realizar un producto del tercer centro de trabajo hasta que tengamos listos un artículo del primer centro de trabajo y dos artículos del segundo centro de trabajo.

El tiempo de ciclo nos indica cuanto tiempo tarda en fabricarse un artículo en un centro de trabajo. El número mínimo que puede establecerse es de 2 minutos, y no hay tiempo máximo, aunque debe ser un número razonable que concuerde con el tiempo que hay entre envíos, que se va a explicar a continuación.

## TIEMPO ENTRE ENVÍOS

Una vez tenemos artículos terminados en el último centro de trabajo (CT3), estos artículos deben ser enviados al cliente o a otro punto logístico, ya que si no nuestro modelo no podría aceptar más producción.

En la práctica deberíamos fijar un número lo suficientemente bajo para que así nos supongan menos costes de transporte, pero también deberíamos tener cuidado con establecer un tiempo muy alto, ya que llevaría a un cuello de botella en el último centro de trabajo, haciendo que la producción sea mucho más baja, ya que se llegaría a la capacidad máxima de nuestro almacén.

La primera celda, que en la hoja del excel es la CB1, es simplemente un número que se fija para ser utilizado en otras fórmulas. En cambio, la celda inferior, que es la CB2, tiene una fórmula Residuo, que también se va a utilizar para establecer un reloj propio para los tres centros de trabajo.

La fórmula residuo consta de los siguientes elementos: número y número divisor. Funciona dándonos el resto de una división, en este caso de CB2 y CB1. Se ha decidido sumar 1 en el primer elemento, ya que, al inicializarse el reloj en cero, si comienza haciendo el resto de 0 dividido entre el reloj de envíos nos daría un error de cálculo.

Fórmula utilizada:

$$=SI(\$G\$2=""; RESIDUO (CB2+1; CB1);0)$$

También hemos tenido en cuenta que solo funcione cuando el valor de la celda G2 está en blanco, que solo ocurre cuando el modelo está funcionando una vez que hemos pulsado la celda de inicio.

Por lo tanto, el valor de la celda CB2, correspondiente al reloj de envíos será igual a 0 si el modelo no está en curso, y cuando esté en curso variará entre 1 y CB1 -1. Para el caso de un tiempo entre envíos de 50, variará entre 1 y 49. La ilustración del reloj es la siguiente:

<b>Tiempo transcurrido entre envíos</b>	<b>50</b>
<b>Reloj de envíos</b>	<b>0</b>

Tabla 19: Celdas donde establecemos el tiempo de ciclo y nos muestra el reloj de los envíos.

## CENTRO DE TRABAJO 1

Dentro de los tres centros de trabajo que tenemos en el modelo de simulación, el primer centro de trabajo se encarga de elaborar un tipo de lote, denominado “P”, que junto con los lotes “Q” del segundo centro de trabajo, nos permitirán confeccionar los artículos “R”, listos para poder ser enviados.

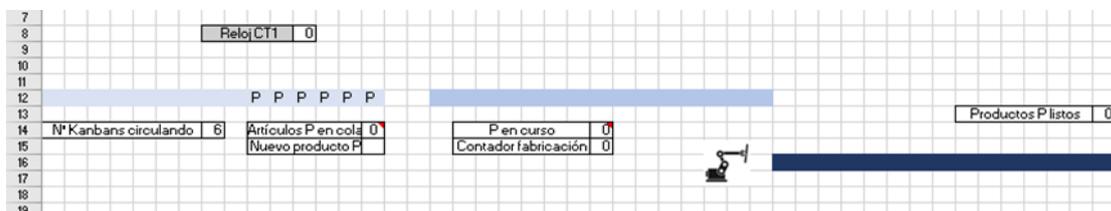


Ilustración 26: Representación del CT1.

En primer lugar, tenemos un reloj de eventos, llamado Reloj CT1, ubicado en la celda L8. Este reloj funciona de forma muy similar al reloj de envíos que hemos explicado anteriormente. También se ha utilizado una fórmula residuo, que es la siguiente:

=SI (G2=""; RESIDUO (L8+1; BU6);0)

Obtenemos el resto entre la celda L8, y el tiempo de ciclo correspondiente al CT1, que se ubica en la celda BU6. Sumamos 1 ya que así inicia el proceso haciendo el resto de 1 dividido entre el tiempo de ciclo correspondiente, para que así no nos dé todo el rato un valor igual a cero.

Como podemos ver en la imagen, tenemos tres zonas bien diferenciadas, que tienen una tonalidad distinta en función de la fase en la que se encuentren. Estas tres zonas son:

- Zona de Cola: Aquí las tarjetas Kanban esperan a que sean admitidas a la zona de fabricación
- Zona de fabricación: En esta zona ya se van elaborando las piezas, en función de una serie de condiciones que veremos más adelante.
- Zona de artículos terminados: Aquí las piezas "P" ya están listas para pasar al centro de trabajo CT3, debido a que ya ha finalizado su producción.

Aparte del reloj también tenemos otros elementos como el número de Kanban que están en circulación. La fórmula utilizada es la siguiente:

=BK6-O14-Y14

Donde BK6 es el número de Kanban que habíamos fijado en un principio, O14 es el número de artículos que tenemos en la cola, e Y14 es el número de productos P que tenemos en curso.

#### Zona de cola:

En esa zona de cola, para cada celda hay una fórmula que va representando la fórmula de los artículos que hay en circulación. Se va llenando de derecha a izquierda en función del número de Kanban que estén en circulación, hasta un máximo de 15, que es la capacidad de la cola.

La fórmula O14, que son los artículos en la cola es una fórmula que lo único que hace es contar el número de artículos que tenemos a lo largo de la zona sombreada de azul claro, que es la zona de cola de los productos.

Justo en la celda inferior, O15, cuando se produce un artículo o pieza P aparece la letra P, lo que servirá para la ejecución de las fórmulas posteriores.

Además, tenemos el **contador de fabricación**, que se encarga de contar si hay ítems P en la zona de fabricación.

Justo encima tenemos además la fórmula de los productos **P en curso**. Esta fórmula es algo más compleja. El código es el siguiente:

```
=SI($G$2="" ;SI(Y(O15="P";($L$8=$BU$6-1);$BH$22<$BK$8;$BU$21<15);$Y$14+1;SI(Y(Y14>0;Y15>=1;Y22>2;$BH$21<$BK$8);$Y$14-1;$Y$14));0)
```

Solo nos da un valor que no sea cero cuando se está ejecutando la simulación, y además tiene que cumplir las siguientes condiciones:

- En la celda de **Nuevo Producto P** aparece una P, que significa que un nuevo ítem pasa a la zona de fabricación, que sucede a su vez cuando se cumple lo siguiente:  
=SI(G2="" ;SI(L8=BU6-1;SI(O12="P";"P";O15);"";""))
  - El reloj del centro de trabajo es igual al tiempo de ciclo menos uno.
  - La primera posición de la zona de cola es igual a P
- El reloj en el Centro de trabajo 1 debe ser igual al tiempo de ciclo menos una unidad, ya que la pieza pasa a la zona de espera un minuto antes de que comience a fabricarse.
- Además, el número de tarjetas Kanban en el centro de trabajo 3 debe ser menor al número especificado en un primer momento, ya que si no se sobrepasaría la capacidad de las zonas de fabricación o de las colas.
- También los productos en curso del tercer centro de trabajo deben ser menor de 15, que como ya sabemos es la capacidad máxima de las colas.

Si se dan estas tres condiciones, sumamos una unidad a los productos P en curso, ya que se acaba de poner en marcha la producción de un ítem.

En el caso de suceder lo contrario, hay varios supuestos por los que restamos una unidad a los productos P en curso o simplemente lo dejamos como estaba. Se reduce su número en una unidad si sucede lo siguiente:

- Los productos P en curso son mayores de la unidad
- El contador de fabricación de P es mayor que 1
- El contador de fabricación de P es mayor que 2
- Los Kanban de R son menores que el número de tarjetas Kanban que habíamos especificado en un principio.

Si esto tampoco se cumple, entonces la celda P en curso no sufre ningún cambio.

#### Zona de fabricación:

Justo debajo de la zona de fabricación, se ha hecho una fórmula auxiliar a la principal, en la que se va representando como desaparecen los Kanban primero de ahí y luego de la zona de fabricación. Por ello, que el reloj debe ser igual al tiempo de ciclo menos

dos unidades, en lugar de restar una unidad, para que desaparezca justo un minuto antes aquí que de la zona de fabricación.

En la hoja de cálculo aparece oculta para que se pueda visualizar mejor la zona de fabricación y no dé lugar a equivocación.

=SI(Y(\$L\$8=(\$BU\$6-2);\$BH\$22<\$BK\$8);AA12;AB13)

Por ejemplo, esta fórmula corresponde a la celda AB13. Si se cumple la condición de que el reloj de CT1 es igual al tiempo de ciclo menos dos unidades y que el número de Kanban del centro de trabajo CT3 es menor que la cantidad que hemos fijado, entonces en esta fórmula aparece el valor de la celda superior izquierda. En caso contrario, se queda como estaba.

Esta fórmula se utiliza para la zona de fabricación ya que dentro de la fórmula de la zona de fabricación se hace referencia a esta fila justo debajo. Se va arrastrando a lo largo de la fila, quedando la celda más a la izquierda vacía, ya que haría referencia a una celda superior izquierda que no tiene ningún valor asignado.

En el caso de la zona de fabricación también se va arrastrando la fórmula de derecha a izquierda, ya que se va llenando la cola de derecha a izquierda. Para una celda cualquiera la fórmula utilizada es la siguiente:

=SI(\$G\$2>"";"";SI(\$L\$8=(\$BU\$6-1);AA13;SI(Y(\$H\$14>5;\$BH\$22<\$BK\$8);"P";AA12)))

Donde la fórmula "\$G\$2>"" evaluará como VERDADERO si la celda G2 no está vacía (contiene algún valor) y como FALSO si la celda G2 está vacía (no contiene ningún valor). Es decir, cuando se esté ejecutando va a funcionar, y cuando se pare o reinicie, esta fórmula dejará de funcionar.

Se tiene que cumplir además que el reloj de fabricación sea igual al tiempo de ciclo menos una unidad. Si eso se cumple, cogemos el valor de la celda auxiliar inferior. En caso contrario, si el número de Kanban en circulación es mayor que la posición de la celda en la cola y el número de Kanban asociados a R es menor que los que habíamos especificado para el tercer centro de trabajo, aparece la letra "P". De no cumplirse ninguna de las anteriores condiciones, no hay ningún cambio en la celda.

#### Zona de artículos terminados:

En esta última zona, los artículos van apareciendo en la zona de artículos terminados, en función de la producción en curso.

Además, hay un contador de productos terminados, en el que con la función de Excel CONTAR.SI lleva a cabo un recuento de la cantidad de artículos P que hay en la zona. El valor de esta celda para el centro de trabajo, junto al valor de esa misma celda para el segundo centro de trabajo, se utilizará para determinar el número de Kanban correspondientes a los productos P o Q en el centro de trabajo CT3.



Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

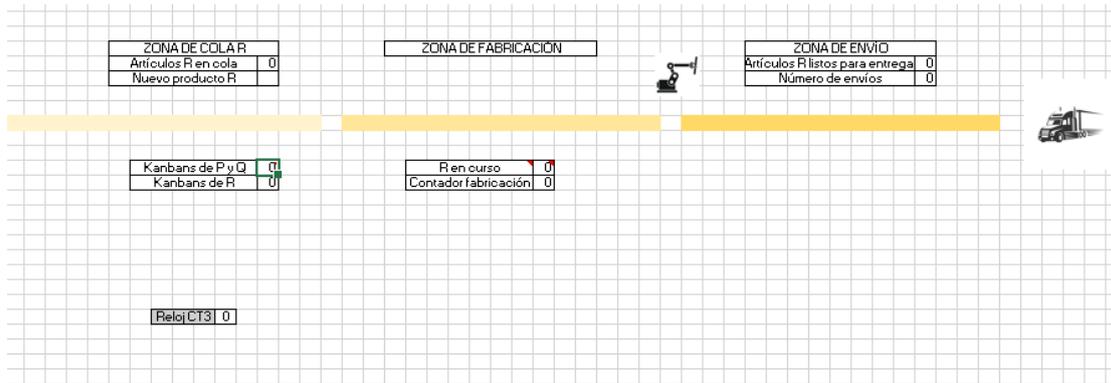


Ilustración 28: Representación CT3.

Para comenzar, el reloj de eventos que tenemos funciona igual que en los dos centros de trabajo anteriores, con una función residuo que recoge el resto de esa misma celda + 1 unidad, dividida entre el tiempo de ciclo o de fabricación que hemos asignado a este centro de trabajo.

Como se ha dicho, en este centro de trabajo tenemos una fórmula correspondiente a los **Kanban procedentes de P y Q**, que tiene en cuenta los artículos que están siendo procesados en estos centros de trabajo.

Tenemos la siguiente fórmula:

=SI(G2="" ;SI(Y(\$AU\$13>1;\$AU\$24>2;\$BH\$21<\$BK\$8);\$BH\$21+1;SI(Y(\$BH\$21=\$BK\$8;\$BZ\$18="" );0;\$BH\$21));0)

Como siempre, la iniciamos con la condición de que la celda G2 esté vacía. Entonces, si hay ya un producto listo de P y dos productos listos de Q, y siempre que esta celda sea menor que los Kanban correspondientes al tercer centro de trabajo, entonces se añade una unidad.

En cambio, si el número de Kanban proveniente de P y Q es igual al número máximo que habíamos establecido en la tabla superior, y el primer artículo en la cola de fabricación de este centro de trabajo es igual a cero, se mantiene el valor de la celda. Si no se cumple ninguna de estas condiciones anteriores, también tiene el mismo valor.

Zona de cola:

También tenemos una fórmula correspondiente a los **Kanban de R**, al igual que en los anteriores centros de trabajo, que simplemente cuenta el número de Kanban que hay en la zona de cola.

Los **artículos R en cola** también tienen una función CONTAR.SI pero en la zona de fabricación de productos, que va justo después de la cola.

La función asociada al **nuevo producto R** funciona muy parecido, por no decir de manera idéntica, a las de los centros de trabajo 1 y 2.

En la zona de cola cada celda tiene asignada una fórmula, que se va arrastrando de derecha a izquierda, al igual que en los casos anteriores como ya se ha explicado. El código es el que viene a continuación, y se ha cogido el ejemplo de la celda que tiene la primera posición en la zona de cola:

=SI(Y(\$B1\$2=1;\$BK\$8>0);"R";SI(\$BH\$21>0;"R";""))

Esto quiere decir que en cuanto se inicia la simulación, es decir, cuando el reloj comienza a funcionar y vale 1, es cuando se inicia poniendo el valor de R. Además, necesita que el número de Kanban que hemos introducido en este centro de trabajo sea superior a 0.

Si no se cumple esta condición, entonces si los Kanban provenientes de P y Q son mayores que un cierto número se va rellenando. Por ejemplo, en la cuarta posición de la cola tendremos que poner como condición necesaria que el número de Kanban sea mayor que 3, para que así solo rellene todas las celdas hasta el número de Kanban provenientes de P y Q. Esto se va arrastrando, hasta que llegamos a la última celda de la zona de cola y solo se rellena si el número de Kanban proveniente de P y de Q es igual a 15.

#### Zona de fabricación:

En el caso de los **artículos R en curso**, se sumará un nuevo producto en curso si se dan los siguientes supuestos:

- El tiempo de fabricación sea igual al tiempo de ciclo menos 1.
- Nuevo producto aparezca con la letra R

En el caso contrario, representando que los artículos se van completando y dejan de estar en curso, se suma resta una unidad a los R en curso, dándose en los siguientes casos:

- Los R en curso sean mayores de 0
- El reloj de envíos sea igual al tiempo transcurrido entre envíos menos uno

Si no se da nada de lo que acabo de comentar, entonces los R en curso mantienen el valor de la celda.

A mayores hay un contador de fabricación, al igual que en CT1 y CT2, que se encarga de contar si están siendo fabricados los productos de la fila de la zona de fabricación, que se va a explicar a continuación:

El funcionamiento es muy similar a como se ha explicado en el primer centro de trabajo, con un centro de trabajo auxiliar justo debajo, que se ha mantenido oculto para que no dificulte la visualización de las piezas que van pasando por esa zona.

Pero las condiciones de nuevo producto en la zona de fabricación sí que han cambiado respecto al CT1 y al CT2:

```
=SI($G$2>"";"";SI($BF$31=($BU$8-1);BZ19;SI(Y($BH$22>0;$BH$22=$BK$8;$BZ$18="");"R";BZ18)))
```

En este caso, solo aparece un nuevo producto cuando la zona de cola está completa como bien podemos ver en el término  $\$BH\$22=\$BK\$8$ , donde BH22 son los Kanban de R y BK8 el número de Kanban que hemos fijado. Si se da esta condición y que esa posición de la cola este vacía, es decir, que no aparezca una R, entonces aparece una letra R en esa posición.

#### Zona de artículos terminados:

Para concluir con la explicación del modelo en este centro de trabajo, en la fila de artículos listos para enviar, se van representando con la letra R según el número de artículos R en curso que tengamos.

También tenemos los artículos listos para ser enviados al cliente, que es simplemente un contador del número de productos en la zona de envíos. Estos lotes van abandonando la zona de envíos de uno en uno, cada vez que se cumple el ciclo del tiempo entre envíos.

El número de envíos por su parte, cada vez que pasa el tiempo entre envíos del cliente suma una unidad. Con todo esto, este apartado referente a la explicación del modelo ya queda finalizado, pasando ya a los resultados de la simulación.

#### 4.2.1 RESULTADOS Y CONCLUSIONES

En los resultados que se van a mostrar a continuación hay muchos parámetros que pueden afectar al resultado de la simulación. Entre ellos el número de tarjetas Kanban, el tiempo de ciclo, el tiempo entre envíos del cliente, etc.

Todo ello mostrará diferentes resultados, que se van a ir explicando con tablas o gráficos. Primero vamos a comenzar con una simulación con los parámetros que habíamos determinado en un principio, y parando el reloj en los en distintos múltiplos de 100 minutos, con un tiempo entre envíos a los clientes de 30 minutos.

Tenemos 6, 9 y 15 Kanban respectivamente para los tres centros de trabajo. En el caso del tiempo de ciclo es de 10, 15 y 20 minutos. En todos los resultados en las

simulaciones se va a parar la misma cuando el reloj marque 1000, que se corresponden a 1000 minutos transcurridos en la línea de producción. Primero se va a mostrar por pantalla los resultados cuando la simulación se encuentra en el instante 1000. Nos vamos a enfocar sobre todo en la productividad de cada centro de trabajo, es decir, el porcentaje de lotes que se producen respecto a una situación ideal, en la que no hay paradas en el proceso de producción. Los resultados son los siguientes:

Centro de trabajo	Nº Kanbans	Tiempo de ciclo	Trabajo en curso	Producción	Producción óptima	Productividad
CT1	6	10	2	75	100	75%
CT2	8	15	1	52	66	79%
CT3	10	20	12	50	50	100%

*Tabla 20: Resultados primera simulación.*

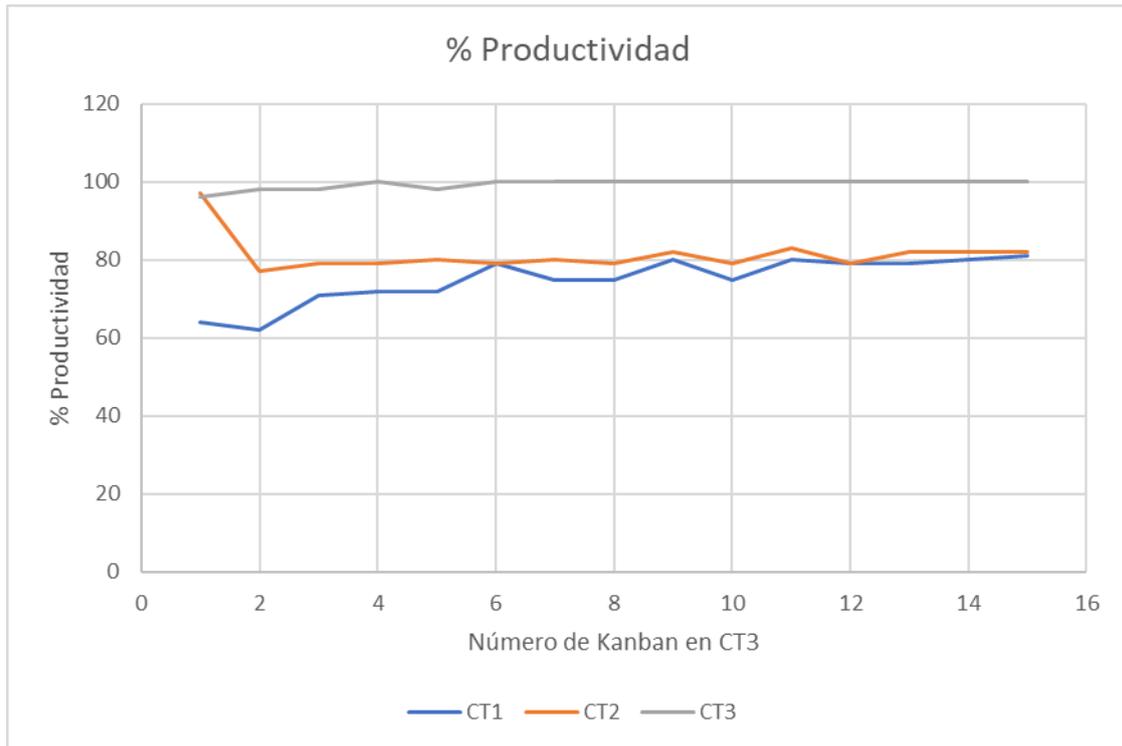
En este caso podemos comprobar que la productividad en los dos primeros centros de trabajo es más baja que en el tercer centro de trabajo, de un 75 y un 79 % en ambos casos, debido a que cuando la cola del tercer centro de trabajo está llena, el modelo ha sido diseñado para que en cuanto eso ocurra, en ese preciso instante dejen de producirse piezas tanto en el primer centro de trabajo como en el segundo.

En cambio, en el tercer trabajo tenemos productividades muy altas, ya que entre otras cosas el tiempo entre envíos no es muy superior al tiempo de ciclo del producto correspondiente al CT3, lo que no hace generar colas ni esperas, aunque podemos comprobar que hay mucho trabajo en curso, algo que a la larga generará retrasos en la producción. En caso de aumentar ese tiempo comenzaría a bajar la productividad en el centro de trabajo 3 drásticamente.

#### Influencia número de tarjetas Kanban:

Un cambio en el número de tarjetas es trascendental como se ha ido explicando a lo largo de todo el trabajo. Su disminución supone un aumento en las interrupciones y una más que probable escasez de productos o piezas. En cambio, si aumentamos este número lo que va a ocurrir es que va a haber mucho trabajo en curso, aparte de llegar a una sobreproducción de lotes, lo que daría lugar a inventarios en grandes cantidades, disparando los costes de almacenaje. Esto podemos comprobarlo de una manera práctica en el modelo preliminar. Se ha hecho un experimento acerca de cómo influyen el número de tarjetas en la productividad y en el trabajo en curso.

Este experimento se va a realizar hasta un rango de 15 tarjetas, ya que es el tamaño máximo de la cola, por lo que no es conveniente utilizar un mayor número, debido a esa limitación de tamaño de la cola. Se va a experimentar la evolución de la productividad para cada uno de los tres centros de trabajo.

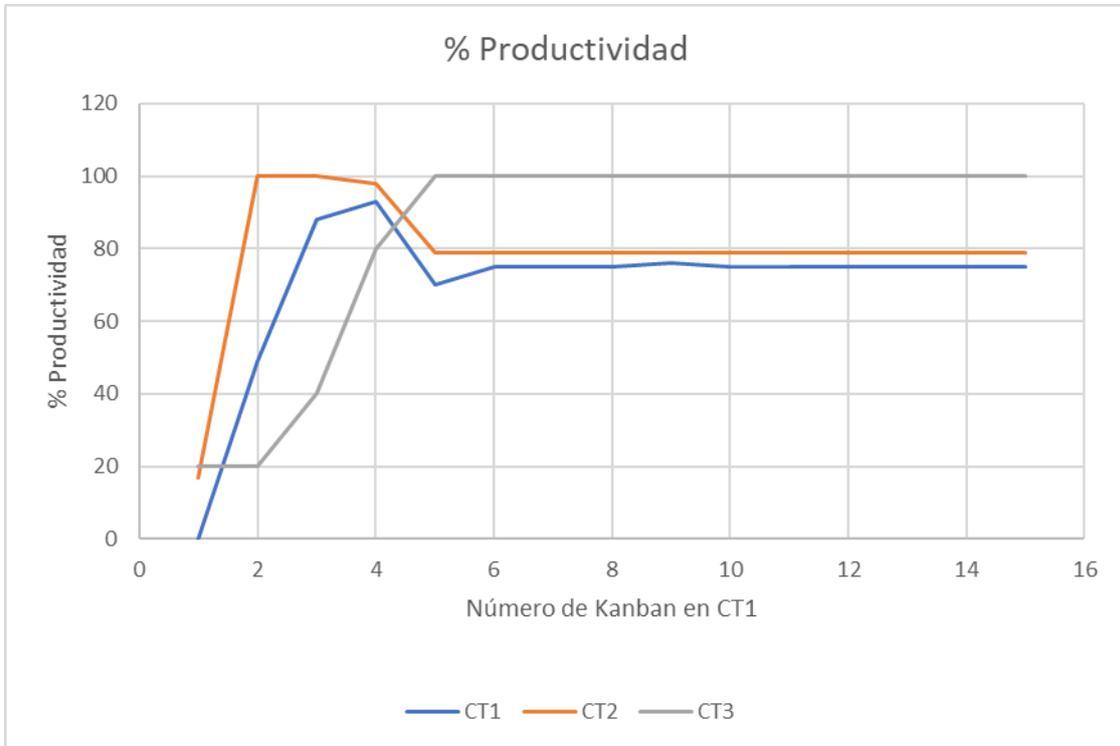


*Ilustración 29: % Productividad de los 3 centros de trabajo en función del número de tarjetas Kanban en CT3.*

En esta gráfica podemos observar que no hay grandes variaciones para la productividad en función del número de tarjetas Kanban, estabilizándose a partir de 3 Kanban en CT3. Sin embargo, sí que es cierto que el porcentaje va aumentando paulatinamente para los dos primeros centros de trabajo, siendo el número óptimo de Kanban de 9 y 11.

Ahora se va a llevar a cabo el mismo proceso para el caso de CT1 y CT2, cambiando el número de tarjetas Kanban para ambos casos.

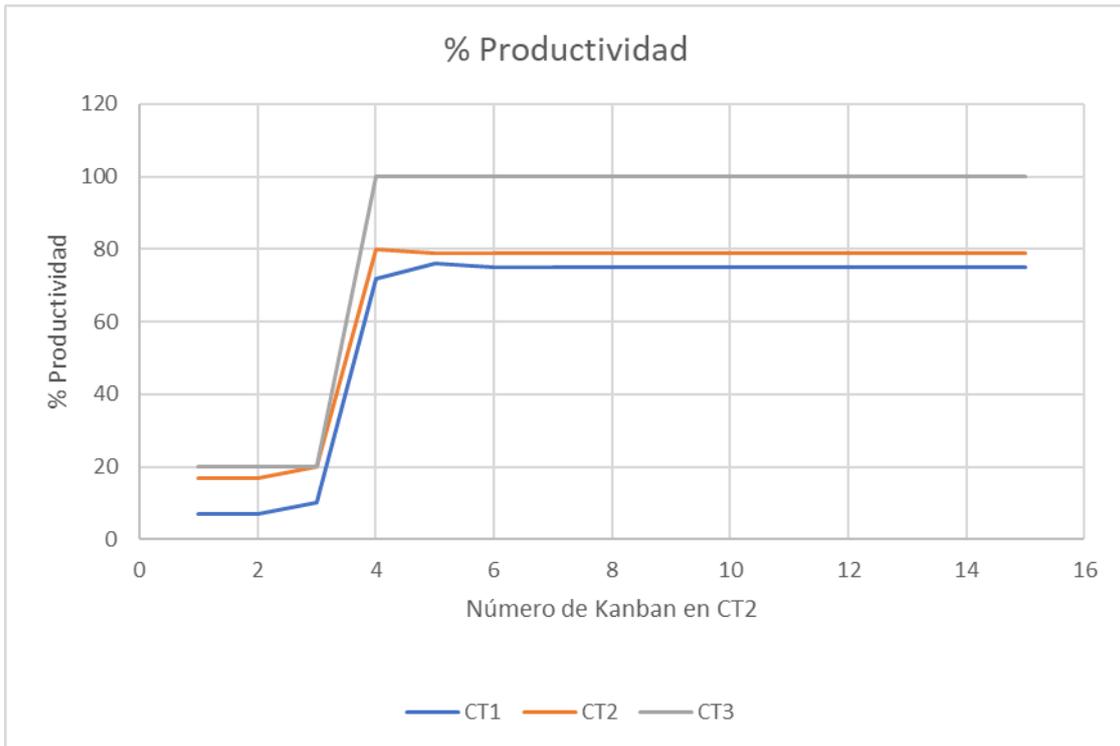
En el caso de CT1 se va a mostrar a continuación la gráfica del % de productividad, fijando el número de Kanban para CT2 en 8 tarjetas y para CT3 en 10 tarjetas.



*Ilustración 30: % Productividad en los 3 centros de trabajo en función del número de tarjetas Kanban en CT1.*

Al igual que para cuando cambiamos el número de tarjetas Kanban en CT3 podemos ver una estabilización a partir de 4 Kanban en el centro de trabajo en el cual estamos variando el número de tarjetas. Hay una gran estabilización, por lo que el número óptimo de tarjetas puede ser de 6 en adelante.

Por último, se va a repetir el proceso cambiando el número de tarjetas en CT2, fijando el número de tarjetas Kanban en 6 y 10 respectivamente para CT1 y CT3.



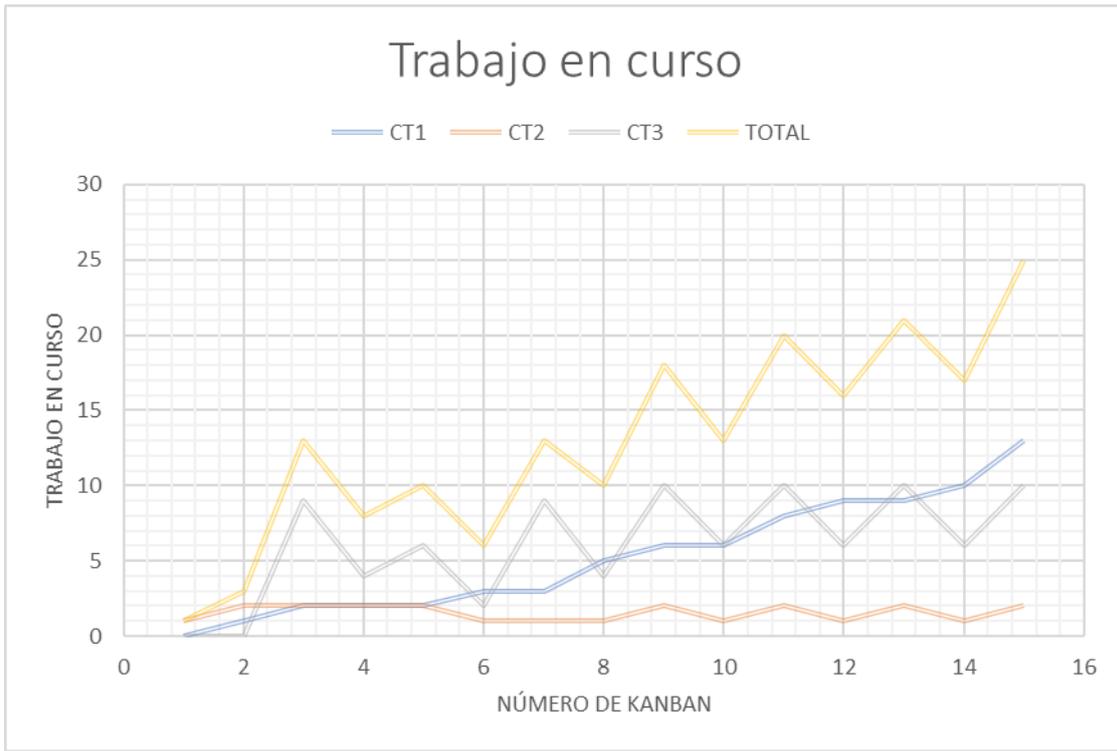
*Ilustración 31: % Productividad en los 3 centros de trabajo en función del número de tarjetas Kanban en CT1.*

Como podemos ver, en este caso se produce un poco más tarde la estabilización de la gráfica que en el caso de CT1. Esto se puede deber debido a la restricción por la cual CT3 solo puede comenzar a producir piezas si hay listas 1 pieza de CT1 y dos piezas de CT2.

En definitiva, acabamos de ver que a partir de cierto número de tarjetas la productividad se mantiene más o menos constante para los tres centros de trabajo, por lo que serían válidas muchas opciones para el número óptimo de tarjetas.

En cambio, debemos tener en cuenta otros factores como el trabajo en curso. Como se han ido apuntando a lo largo del trabajo, el trabajo en curso o WIP en inglés aumenta a medida que aumentamos el número de tarjetas Kanban y viceversa. Es importante que el trabajo en curso sea bajo, ya que su acumulación puede derivar en retrasos y cancelaciones de producción.

Debido a su importancia se ha desarrollado un gráfico en el que se puede ver que efectivamente el trabajo en curso va aumentando a medida que aumentamos el número de tarjetas Kanban.



*Ilustración 32: Trabajo en curso en función del número de tarjetas Kanban.*

Sin embargo, cabe destacar que estos datos se han cogido en un momento determinado de la simulación, en este caso en el minuto 1000 de la misma. Aunque veamos que como era de esperar va aumentando, el resultado puede llevar a engaño debido a que no representa el trabajo en curso a lo largo de la simulación, si no en un determinado instante.

A pesar de ello me ha parecido interesante incluirlo debido a que ratifica lo que se ha indicado a continuación al ver la tendencia ascendente de la gráfica. Cuanto mayor número de tarjetas Kanban haya en circulación habrá una mayor cantidad de trabajo en curso.

#### Influencia tiempo entre envíos:

El hecho de modificar cada cuanto tiempo enviamos productos al cliente puede determinar de gran manera la productividad, siendo vital en este modelo. Esto se debe a que con un tiempo muy alto se producirán cuellos de botella y con un tiempo muy bajo entre envíos, supondrían en la realidad un mayor coste para la línea. Se va a demostrar su influencia en la productividad del último centro de trabajo, modificando esa celda de tiempo entre envíos, mostrando primero los parámetros utilizados en dichas simulaciones.

Centro de trabajo	N.º Kanban	Tiempo de ciclo
CT1	6	10
CT2	8	15
CT3	10	20

Tabla 21: Parámetros utilizados en la simulación del modelo preliminar.

Con estos parámetros se va a ir modificando el tiempo entre envíos para validar la hipótesis de que a mayor tiempo entre envíos se produce una reducción en la productividad del último centro de trabajo, que es el centro principal ya que en él se ensamblan las piezas y su resultado final es el que llega al cliente.

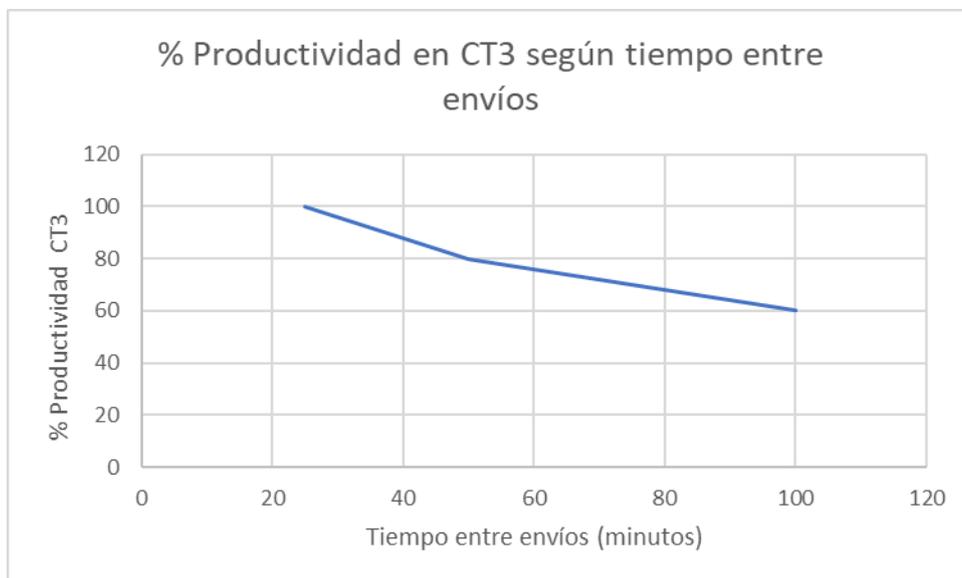


Ilustración 33: % Productividad en CT3 en función del tiempo entre envíos.

En esta gráfica se puede comprobar de primera mano cómo va reduciéndose la productividad a medida que aumentamos el tiempo entre envíos. Esto se ve con más claridad a medida que avanza la simulación y se empieza a saturar el tercer centro de trabajo, debido a que se producen lotes a un ritmo más alto del que salen del modelo.

Con un tiempo entre envíos de 25 minutos la productividad es del 100%, pero en cuanto cambiamos a 50 y sobre todo a 100 ya cae en picado. Como bien se ha dicho antes, habría que optimizar este tiempo entre envíos para así poder tener una productividad alta pero no tener muchos costes asociados a estos envíos, ya que cuanto menos tiempo haya entre envíos, mayores costes de transporte va a tener que soportar la fábrica.

## CONCLUSIONES

En este modelo preliminar hemos ido demostrando mediante simulaciones como influye en el rendimiento de la línea de producción. En una fábrica hay infinidad de factores que pueden afectar a la producción, pero en mi caso se ha decidido hacer hincapié en los más relevantes y los que tienen que ver más con este TFG.

Como resumen, se ha demostrado la influencia del número de tarjetas Kanban en el trabajo en curso y en la productividad en los tres centros de trabajo. Por último, hemos hecho una pequeña referencia a la influencia en el tiempo entre envíos en el funcionamiento de la línea de producción.

## 4.3 MEJORA DEL MODELO

A partir del modelo de simulación que acabo de explicar, una posible mejora que se podría hacer es la implementación de una matriz de flujo en la que escogiéramos qué Centro de Trabajo requiere de piezas de los anteriores centros de trabajo.

Podríamos poner 2 centros de producción y 2 centros de ensamblaje. Para determinar qué centro de trabajo requiere de producción y cual no, realizaríamos una matriz 2x2 en la que si hay un 0 no requiere piezas, pero si tenemos un 1 sí que requiere de ellas. Este modelo podría ampliarse a 3x3 a 4x4, etc., pero se ha decidido fijarlo en 2x2 para así visualizarlo de la forma más sencilla posible.

Los centros de producción que surten de piezas a los centros de ensamblaje serían CT1 y CT2

Los centros de ensamblaje por su parte serían CT3 y CT4

MATRIZ	CT3	CT4
CT1	0	1
CT2	1	1

*Tabla 22: Ejemplo de matriz de flujo.*

- En este caso, CT3 requeriría piezas solamente de CT2
- CT4 necesitaría piezas de CT1 y CT2

Gráficamente, ahora tenemos un “layout” distinto también, ya que pasaríamos de tener 3 centros de trabajo a tener 4.

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

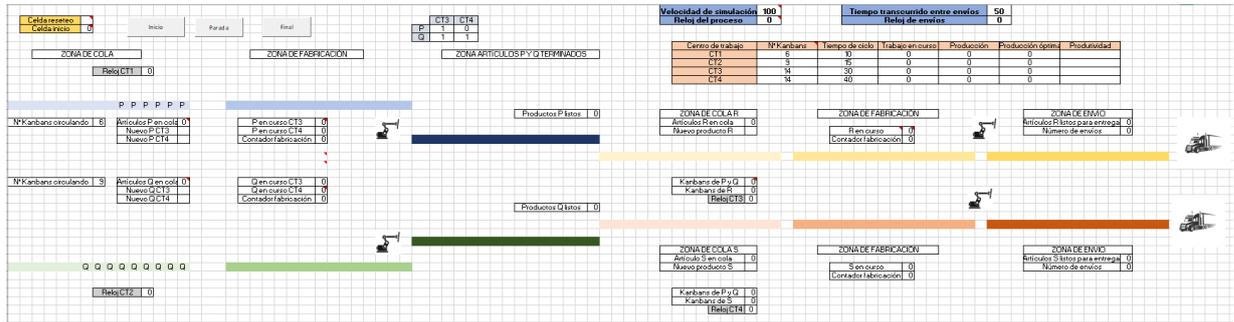


Ilustración 34: Representación modelo mejorado.

MODIFICACIONES PERTINENTES

Visualmente se parece mucho a la simulación inicial, con dos diferencias que se aprecian muy fácilmente:

- Como es lógico ahora aparece un centro de trabajo más, ya que ahora al tener cuatro centros de trabajo y antes tres, ese centro de trabajo adicional era necesario representarlo.
- Además, aparece la matriz que hemos mencionado justo antes, a partir de la cual los centros de trabajo 3 y 4 van a necesitar piezas de ningún centro de trabajo, solo de P, solo de Q, o de los dos simultáneamente.

Todo ello va a hacer que haya tenido que hacer cambios en las fórmulas, debido a la exigencia por ejemplo de que tengamos listas piezas de los dos centros de trabajo. También se ha debido tener en cuenta que no siempre van a ser necesarias piezas del centro de trabajo 1 y del 2, si no que puede ser de uno de los dos solamente o de ninguno.

Las fórmulas del centro de trabajo 4 son muy parecidas a las del centro de trabajo 3, realizando unas pequeñas modificaciones.

Las celdas del modelo correspondientes a cada flujo son las siguientes:

MATRIZ	CT3 (R)	CT4 (S)
CT1 (P)	AH3	AJ3
CT2 (Q)	AH4	AJ4

Tabla 23: Celdas correspondientes a cada elemento de la matriz de flujo.

## CENTRO DE TRABAJO 1:

### Zona de cola

En este centro de trabajo se ha tenido que distinguir entre las piezas que van a ir a parar al CT3 y las que se dirigen al CT4, teniendo una nueva fórmula para la producción de piezas para CT3 y para CT4, **Nuevo P CT3:**

=SI(Y(G2="" ; AH3=1); SI(L8=BP6-1; SI(O12="P"; "P"; O15); ""); "")

En esta fórmula hemos tenido que poner la restricción asociada a la matriz (AH3=1). Es igual para los productos P que se dirigen a CT4, pero cambiando la celda a la celda que nos referimos de la matriz a (AJ3=1).

En el caso de los productos que se están fabricando también se ha hecho una distinción entre los productos P que van a parar a CT3 y los que van a pasar a CT4. Simplemente se ha modificado la fórmula igualando a la celda de la matriz que corresponde, es decir, que solo funcione si su casilla correspondiente tiene marcado un 1.

### Fórmula P en curso CT4:

=SI(Y(\$G\$2="" ; \$AJ\$3=1); SI(Y(O16="P"; (\$L\$8=\$BP\$6-1); \$BH\$35<\$BK\$9; \$BU\$31<15; Y19<15); \$Y\$19+1; SI(Y(Y19>0; Y16>=1; Y22>2; \$BH\$34<\$BK\$9); \$Y\$19-1; \$Y\$19)); 0)

### Zona de fabricación

Al haber ya condiciones iniciales de funcionamiento o no funcionamiento aquí apenas se han tenido que hacer cambios.

### Zona artículos terminados:

En la zona de producción, en cuanto hay un producto P listo ya sea para CT3 o para CT4 aparece la letra "P" en la cola. Esto se debe a que no hay distinción en la cola, van todos los productos P ya vayan a CT3 o a CT4.

## CENTRO DE TRABAJO 2

Al igual que en el anterior modelo, el segundo centro de trabajo es muy similar al Centro de trabajo 1, modificando únicamente los parámetros que le caracterizan respecto al primero.

En este caso también hay distinción entre los productos que van a parar a CT3 o a CT4, tanto los que hay en curso como para la notificación de nuevo producto Q ya vaya a parar a CT3 o CT4.

### CENTRO DE TRABAJO 3

En este centro de trabajo se terminan de producir las piezas ya provengan de CT1 o de CT2. Se han tenido que hacer varias modificaciones respecto al modelo anterior ya que hay que tener en cuenta los valores de la matriz que podemos ver a continuación:

	CT3	CT4
P	1	1
Q	0	1

Tabla 24: Matriz de flujo en la hoja de Excel.

#### Zona de cola:

Para los productos que se van almacenando en la zona de color más claro, que es la zona de cola, se han introducido la condición de que solo funcione cuando CT3 demande productos de al menos de un centro de trabajo, ya sea CT1, o CT2 o ambos, siendo la nueva fórmula la siguiente:

=SI(Y(\$BI\$2=1;\$BK\$8>1;O(\$AH\$3<>0;\$AH\$4<>0));"R";SI(Y(\$BH\$21>0;O(\$AH\$3<>0;\$AH\$4<>0));"R";""))

, en la que con que uno de los dos sea distinto de cero, ya funciona perfectamente.

Para los **Kanban de P y Q**, se han introducido los 4 casos posibles que puede haber en las posiciones de las casillas referidas al CT3:

=SI(G2="";

SI(Y(AH3=1;AH4=1);SI(Y(AU13>1;\$AU\$24>2;\$BH\$21<\$BK\$8);\$BH\$21+1;SI(Y(\$BH\$21=\$BK\$8;\$BZ\$18="");0;\$BH\$21));

SI(Y(AH3=1;AH4=0);SI(Y(AU13>1;\$BH\$21<\$BK\$8);\$BH\$21+1;SI(Y(\$BH\$21=\$BK\$8;\$BZ\$18="");0;\$BH\$21));

SI(Y(AH3=0;AH4=1);SI(Y(\$AU\$24>2;\$BH\$21<\$BK\$8);\$BH\$21+1;SI(Y(\$BH\$21=\$BK\$8;\$BZ\$18="");0;\$BH\$21));

SI(Y(AH3=0;AH4=0);0;0));0)

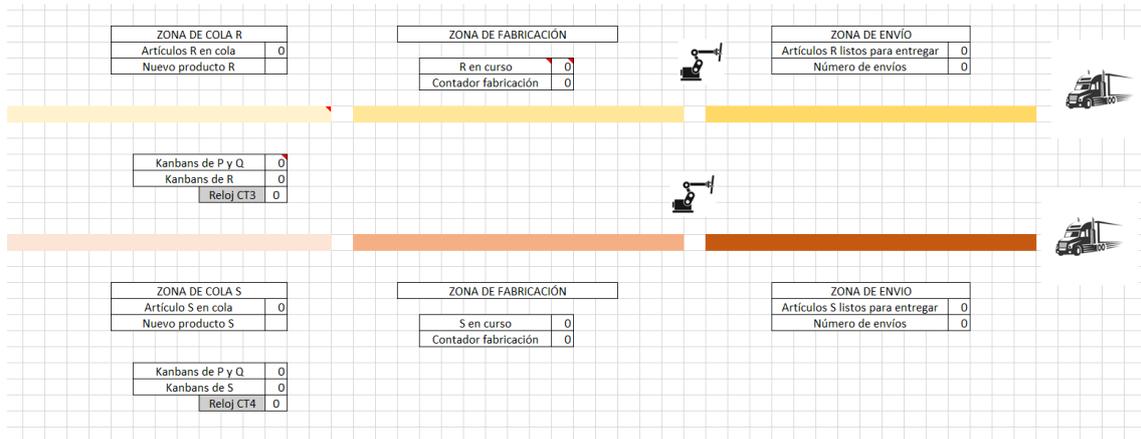
En esta zona del centro de trabajo se han mantenido las fórmulas anteriores

#### Zona de fabricación:

Los R en curso tienen la misma fórmula con la excepción de que solo funcionen cuando uno de los dos valores de la matriz asociados a CT3 sea igual a 1.

Zona de envío:

En esta zona solo ha habido cambios en el número de envíos, que solo ocurren con las condiciones ya nombradas. En la siguiente ilustración se muestran el tercer y el cuarto centro de trabajo



*Ilustración 35: Representación CT3 y CT4.*

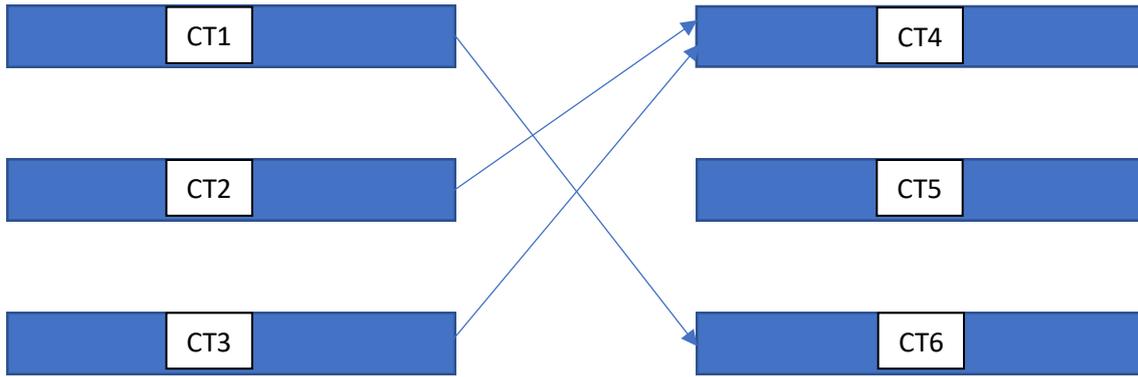
**CENTRO DE TRABAJO 4**

La estructura es la misma que en CT3, teniendo en cuenta los posibles cambios en función de la necesidad de productos. Aquí se incorpora una nueva letra ya que este centro de trabajo es nuevo respecto al modelo original. El flujo de productos se representa con la letra “S”.

**4.3.1 RESULTADOS Y CONCLUSIONES**

Para la conclusión de este apartado, se va a comentar los diferentes resultados que pueden darse en el modelo. Con todo lo explicado anteriormente podremos hacer múltiples combinaciones con los cuatro centros de trabajo, aunque como se ha dicho al principio, esto podría implementarse para un tamaño 3x3, 4x4, etc.

En la ilustración que se inserta a continuación se muestra un ejemplo de cómo funcionaría una matriz de flujo con tres centros de trabajo que proveen de piezas a otros tres centros de trabajo. También se podría ampliar a más centros de trabajo, siendo una posible mejora futura del modelo.



*Ilustración 36: Representación posible distribución en una línea con 6 centros de trabajo.*

La imagen insertada anteriormente corresponde a una posible combinación en una línea de producción 3x3. En este caso CT1 suministra de piezas a CT6, CT2 y CT3 suministran ambos a CT4 y el CT5 queda sin utilizar, ya que no tiene suministro entrante de piezas.

En definitiva, en el modelo que se ha llevado a cabo, que se trata de un modelo 2x2 como ya se ha explicado, se va a mostrar varias combinaciones posibles y los resultados que se muestran por pantalla.

#### PRIMER CASO

A pesar de haber pulsado el botón de inicio, en el supuesto caso de que todos los elementos de la matriz sean cero, el sistema no funcionará y no se verá ningún flujo de materiales. Además, con este sistema, se pueden llevar a cabo experimentos en los que por algún motivo podamos aislar un centro de trabajo cualquiera y comprobar el funcionamiento del sistema, como puede ser el caso del primer centro de trabajo no se encuentre operativo, por lo que el segundo centro de trabajo (piezas Q), suministrará de piezas a CT3 y CT4:

	CT3	CT4
P	0	0
Q	1	1

*Tabla 25: Representación de la matriz de flujo aislando a CT1.*

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

Con esta configuración solo habrá flujo entrante de piezas Q para poder tener los artículos terminados R y S correspondientes a CT3 y CT4. Una vez empieza la simulación podemos observar cómo solo se producen artículos gracias a las piezas del segundo centro de trabajo. Los resultados una vez han pasado 1000 minutos de la simulación son los que se van a mostrar a continuación:

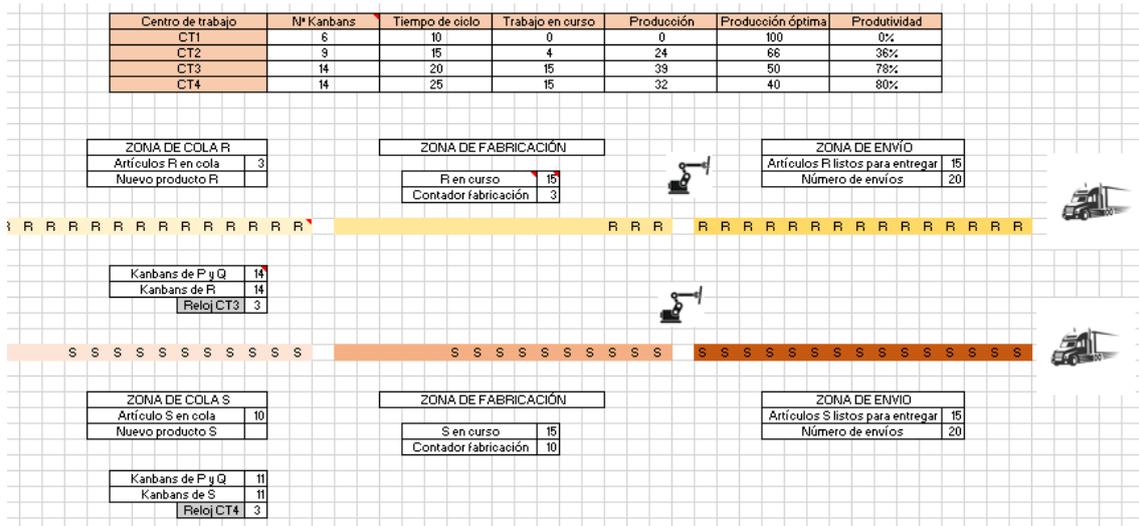


Ilustración 37: Representación del modelo aislando a CT1.

Se ha comprobado que al solo haber productos de un centro de trabajo para CT3 y CT4, era necesario introducir un reloj auxiliar para evitar esos cuellos de botella. Esto se produce debido a que, en el modelo original, requería que hubiera un artículo P terminado y dos artículos Q terminados.

Una vez anulamos uno de estos centros de trabajo, debemos quitar esa restricción ya que se deja de producir en ambos centros y el modelo no funcionaría. Por ello se ha tenido que sustituirla por ese reloj auxiliar según sea el caso.

En cuanto a los resultados, los dos últimos centros de trabajo gozan de una productividad importante, mientras que el CT2, que es el encargado del suministro de piezas, tiene un rendimiento peor. Esto se debe a que a pesar de que se ha introducido un reloj auxiliar hay cuellos de botella, pero esta es la mejor solución para poder visualizar mejor los resultados y ver el flujo de materiales.

SEGUNDO CASO

Una vez se ha hecho este experimento, ahora se va a comprobar el funcionamiento del sistema en el caso de que solo un centro de trabajo de los dos requiera piezas.

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

En este caso CT3 no va a requerir de ninguna pieza y CT4 necesitará piezas de CT1 y CT2, con esta configuración de la matriz:

	CT3	CT4
P	0	1
Q	0	1

Tabla 26: Configuración de la matriz de flujo aislando a CT3.

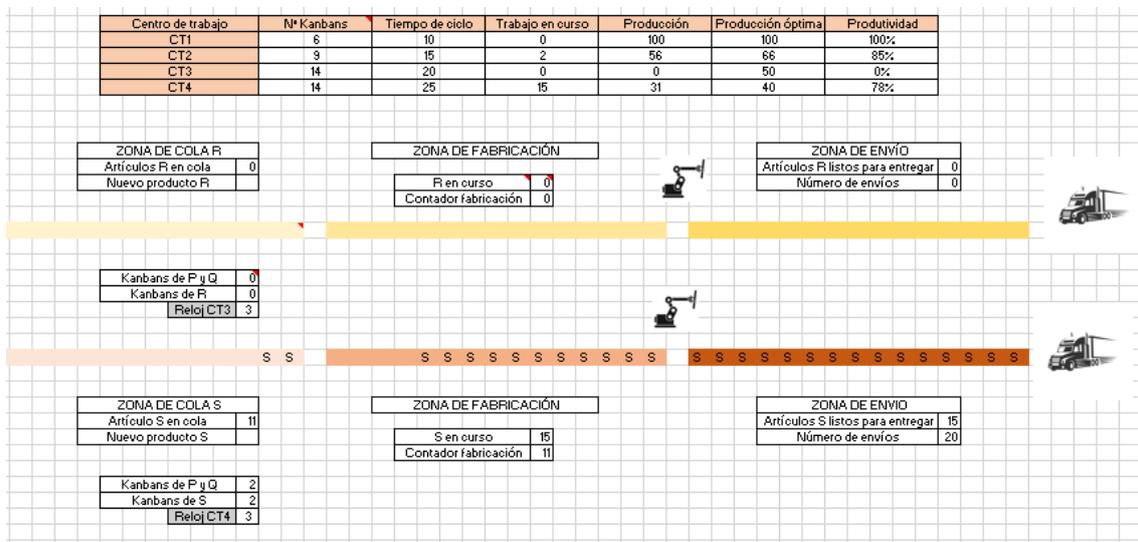


Ilustración 38: Representación del modelo aislando a CT3.

Como podemos comprobar tanto en el flujo de piezas por las zonas de cola y producción como por la producción en los centros de trabajo, se comprueba que el sistema se comporta igual de bien teniendo solamente un centro de trabajo (en este caso CT4) de receptor de piezas.

### TERCER CASO

Por último, se va a mostrar el funcionamiento del sistema cuando están todos los centros de trabajo en funcionamiento, es decir, ambos CT3 y CT4 requieren piezas tanto de CT1 como de CT2. Para ello vamos a tener una matriz de unos como se ve en la siguiente tabla:

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

	CT3	CT4
P	1	1
Q	1	1

Tabla 27: Configuración final de la matriz de flujo.

En este último supuesto se va a poder comprobar cómo hay flujo de materiales en los dos últimos centros de trabajo, como en la imagen que viene a continuación:

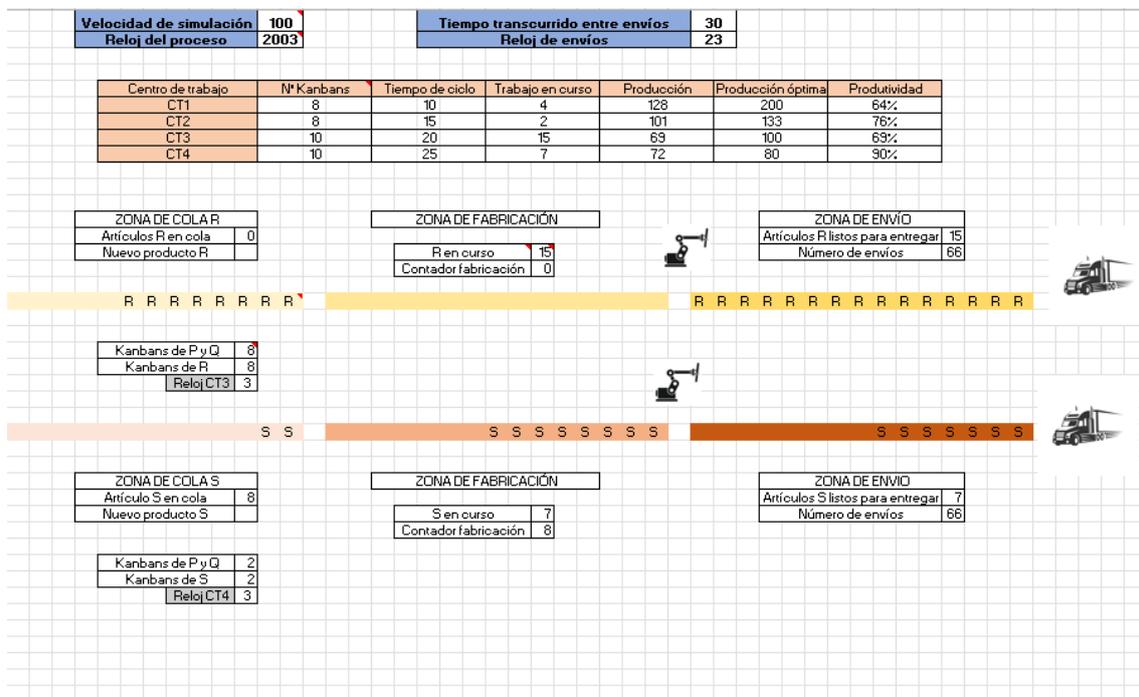


Ilustración 39: Representación del modelo con todos los trabajos en funcionamiento.

Como podemos observar, pasa algo muy parecido al modelo preliminar, en el que los centros de suministro de piezas tienen un menor rendimiento respecto a los dos últimos centros de trabajo, CT3 y CT4.

Esto se debe a que tanto CT3 como CT4 tienen un tiempo de ciclo mayor, por lo que las piezas en CT1 y CT2 se producen de una manera mucho más rápida. Además, solo se producen piezas en CT1 y CT2 cuando uno de los dos últimos centros de trabajo tenga espacio suficiente.

### INFLUENCIA NÚMERO DE TARJETAS

A continuación, se va a realizar una serie de experimentos al igual que en el modelo preliminar para esta configuración de la línea. En un principio se han escogido los siguientes valores para los parámetros en las mismas:

Centro de trabajo	Nº Kanbans	Tiempo de ciclo
CT1	8	10
CT2	8	15
CT3	10	20
CT4	10	25

Tabla 28: Valores utilizados en la experimentación en el modelo mejorado.

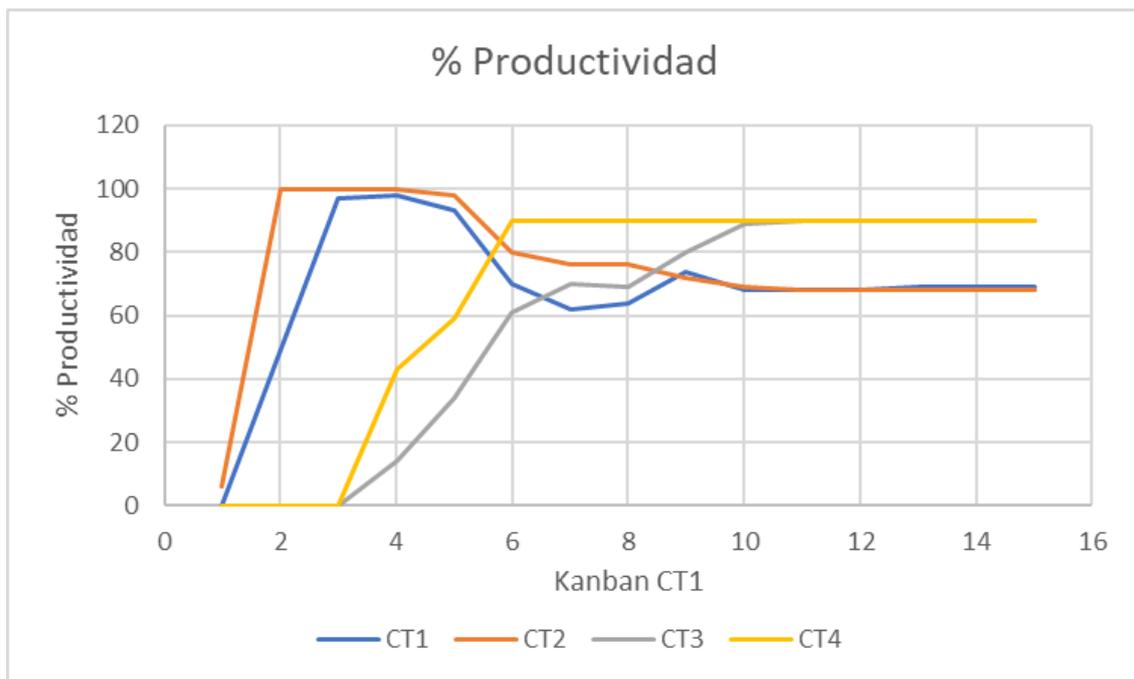


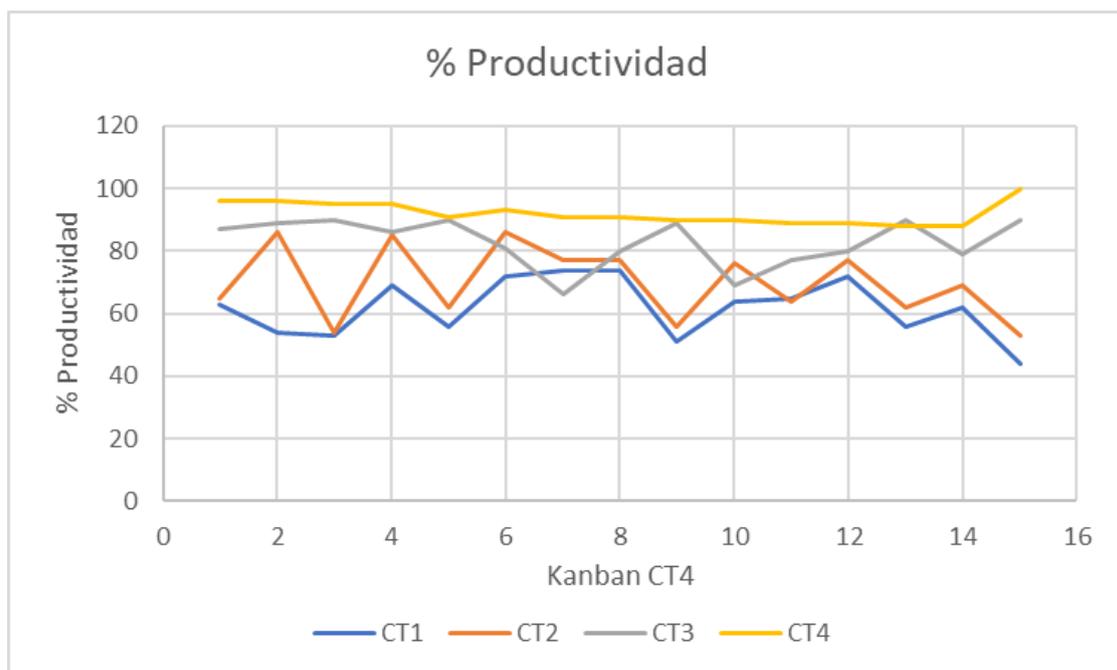
Ilustración 40: % de Productividad en los distintos centros de trabajo en función del número de Kanban en CT1.

Como se ha indicado, se han fijado los valores de los Kanban en CT2, CT3 y CT4 en 8,10 y 10 Kanban respectivamente, y se han ido cambiando el número de Kanban en el primer centro de trabajo.

En esta gráfica se puede observar claramente que cuando tenemos un número de Kanban muy pequeño en uno de los centros que suministran piezas, se producen retrasos y aumentos de los tiempos de espera debido a la escasez de piezas, lo que se puede plasmar en el rendimiento de los centros de trabajo 3 y 4, que van aumentando su productividad paulatinamente a medida que aumentamos el número de Kanban.

El número de Kanban óptimo podría ser cualquiera de los valores entre 10 y 15, ya que se produce una estabilización en el rendimiento para todos los centros de trabajo.

En el próximo gráfico se va a representar la variación de la productividad variando el número de Kanban en CT4, manteniendo constante los números de tarjetas Kanban para CT1, CT2 y CT3.



*Ilustración 41: % de Productividad en función del número de tarjetas Kanban en CT4.*

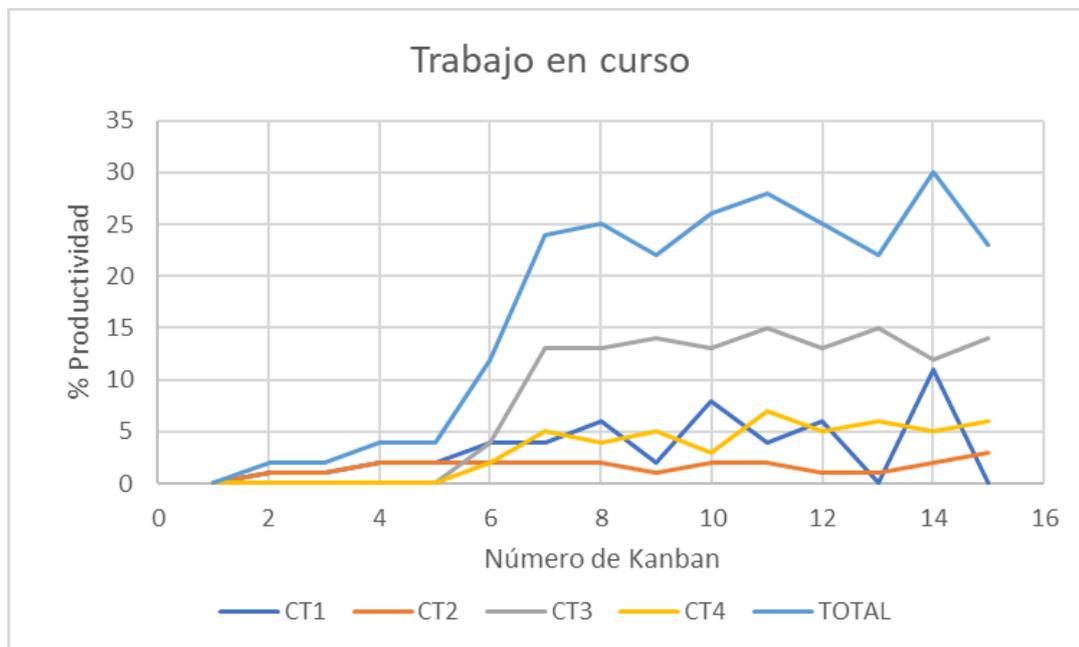
En este caso podemos ver que no se produce una estabilización tan acusada para un número de Kanban superior, como ocurría en el caso de ir cambiando el número de Kanban en CT1.

Se puede comprobar que el cambio de número de Kanban en uno de los centros que reciben piezas produce cambios más acusados en la productividad que en el caso

de que se cambie el número de Kanban en uno de los centros de trabajo que suministran piezas.

El número óptimo de Kanban no se puede determinar de una manera tan clara como en la anterior simulación, pero depende de en qué centros de trabajo queramos optimizar la productividad, como se observa en el gráfico.

Por último, se va a terminar con la evolución del trabajo en curso en los cuatro centros de trabajo en función del número de tarjetas. Como bien se puede observar en la siguiente gráfica, el trabajo en curso va en aumento a medida que aumentamos el número de tarjetas, teniendo un pico para 14 tarjetas Kanban en cada centro de trabajo.



*Ilustración 42: Evolución del trabajo en curso en función del número de tarjetas Kanban.*

Lo ideal es tener un trabajo en curso lo más bajo posible, por lo que es conveniente buscar un número de Kanban que se ajuste al máximo a esta premisa manteniendo un porcentaje de productividad aceptable.

## CONCLUSIONES

En este apartado de la mejora del modelo se ha partido del modelo preliminar y se han hecho modificaciones para adaptarlo a un total de cuatro centros de trabajo. A continuación, se ha expuesto tres supuestos posibles en los que en función de los valores que se introduzcan en la matriz tiene un comportamiento totalmente distinto.

Por último, se ha llevado a cabo una discusión de resultados acerca de las posibles combinaciones de las que acabo de hablar, con el objetivo principal de exponer la influencia en el número de tarjetas tanto para la productividad como para el trabajo en curso en el modelo mejorado.

Determinación del número de tarjetas en circulación en la metodología Kanban.  
Fernando Doral Yagüe

## 5. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo de fin de grado (TFG) se ha mostrado toda la información relevante relativa al sistema Kanban, primero con una breve introducción y los orígenes de esta metodología. En este apartado se han explicado dentro de la metodología Kanban los distintos tipos de sistemas que hay, ya sean de tarjeta simple o de tarjeta doble. Además, se ha explicado los pilares básicos sobre los que se forma este sistema, como los cinco ceros, el Kaizen o las 3 “mus”. También se ha hecho un gran repaso a lo largo de las diferentes técnicas o herramientas sobre las que se han investigado para calcular el número de tarjetas y el tamaño de lote óptimos. En cuanto a esas herramientas nombradas a lo largo del trabajo, todas ellas ya sean metaheurísticas, de métodos exactos, etc. necesitan de una simulación pertinente ya sea con Arena, eKanban o algún otro método posible, para poder comprobar que las conclusiones obtenidas concuerdan con los resultados de dicha simulación. Para ello se ha considerado imprescindible que para hablar de la metodología Kanban es necesario realizar una pequeña simulación para poder ir interpretando los resultados obtenidos y el comportamiento del sistema.

A raíz de la simulación en Microsoft Excel con la ayuda de las macros sobre un sistema Kanban, se ha comprobado la influencia de distintos parámetros en esa línea de producción y la determinación de las mejores cantidades posibles de tarjetas Kanban para cada centro de trabajo, en el que cada tarjeta Kanban representa un lote o pieza en circulación en el modelo.

El modelo de simulación creado supone una línea de producción sencilla, en la que hay un tipo de producto para cada centro de trabajo. A partir de ello, se ha realizado una mejora que hace el modelo más flexible, pudiendo ser extensible hasta un número razonable de centros de trabajo. En ambos modelos se han introducido gráficos e ilustraciones en los que se puede demostrar la influencia del número de tarjetas Kanban. Se ha podido comprobar que su reducción en exceso puede producir retrasos en la producción debido a la falta de suministro de piezas, provocando interrupciones.

En el caso contrario de ser un número muy alto, aumenta el trabajo en curso como se puede observar en la Ilustración 32 del modelo preliminar, donde a medida que aumenta el número de tarjetas se puede observar una tendencia ascendente en el trabajo en curso en todos y cada uno de los centros de trabajo.

En definitiva, en este trabajo se ha intentado abordar desde distintos puntos de vista e información documentada todo lo relativo al sistema Kanban y a su utilización en la actualidad, ya que en su día revolucionó la manera de dirigir una fábrica.

## **5.1 TRABAJOS FUTUROS**

En el modelo preliminar se ha explicado el funcionamiento de la línea de producción y se ha implementado una mejora para hacer el modelo más flexible. Se ha expuesto un modelo con cuatro centros de trabajo, aunque es mejorable introduciendo más centros de trabajo que puedan surtir piezas a los centros de trabajo finales.

Por ello, una posible mejora sería a partir de la introducción de más centros de trabajo, la posibilidad de poder producir lotes de piezas distintos en cada centro de trabajo, y no solamente el mismo tipo de lote para cada centro de producción.

## 6. BIBLIOGRAFÍA

Esparrago Jr, R. A. (1988). Kanban. *Production and Inventory Management Journal*, 29(1), 6.

Peredo, Óscar (2006). *Reconocimiento de patrones de simulación Geoestadística*, 2018.

Shahabudeen, P., Gopinath, R., & Krishnaiah, K. (2002). Design of bi-criteria kanban system using simulated annealing technique. *Computers & Industrial Engineering*, 41(4), 355-370.

Priyom Dutta, B.S. *Cognitive Big Data Intelligence with a Metaheuristic Approach*, 2022

Rabbani, M., Layegh, J., & Ebrahim, R. M. (2009). *Advances in engineering software*, 40(6), 431-437.

Hou, T. H. T., & Hu, W. C. (2011). An integrated MOGA approach to determine the Pareto-optimal kanban number and size for a JIT system. *Expert Systems with Applications*, 38(5), 5912-5918.

Matzka, J., Di Mascolo, M., & Furmans, K. (2012). Buffer sizing of a Heijunka Kanban system. *Journal of Intelligent Manufacturing*, 23(1).

Brar, G. S., & Saini, G. (2011, July). Milk run logistics: literature review and directions. In *Proceedings of the world congress on engineering* (Vol. 1, pp. 6-8). WCE.

Al-Tahat, M. D., Dalalah, D., & Barghash, M. A. (2012). Dynamic programming model for multi-stage single-product Kanban-controlled serial production line. *Journal of Intelligent Manufacturing*, 23, 37-48.

Moran, T. J., & Brayer, K. (2011). A Simulation Study Of Economic Production Quantity Lot Size To Kanban For A Single Line Production System Under Various Setup Times With Annual Setup Cost As Performance Metric. *International Journal of Management & Information Systems (IJMIS)*, 15(2), 23-30.

Socconini, L. (2019). *Lean manufacturing. Paso a paso*. Marge books.

Tapias, Y. A. A., & Correa, J. H. R. (2010). Kaizen: Un caso de estudio. *Scientia et Technica*, 16(45), 59-64.

Sendil Kumar, C., & Panneerselvam, R. (2007). Literature review of JIT-KANBAN system. *The International Journal of Advanced Manufacturing Technology*, 32, 393-408.