



## DESARROLLO DE UN SIMULADOR DE SISTEMAS DE MANUFACTURA FLEXIBLE CON INTERFAZ GRÁFICA BASADO EN REDES DE PETRI

EDWIN LEONARDO GARCÍA AUCATOMA, LUIS FERNANDO ESCOBAR CARVAJAL

Universidad de las Fuerzas Armadas ESPE  
Departamento de Ciencias de la Energía y Mecánica  
Quito, Ecuador

*(Recibido 31 de octubre de 2017, para publicación 23 de enero de 2018)*

**Resumen** – Esta investigación desarrolla ideas en la aplicación de modelos de Sistemas Flexibles de Manufactura (FMS) que pueden ser simulados como sistemas de eventos discretos (SED) y tienen por finalidad identificar a sistemas en los que los eventos que cambian el estado del mismo ocurren en instantes específicos en el tiempo. Los SED pueden modelar muchos de los fenómenos a los que se enfrentan los procesos productivos de una empresa por lo cual se desarrolló un simulador con una interfaz gráfica programada en lenguaje java sustentado en redes de Petri para representar la dinámica de los sistemas y usarlo como una herramienta para la incidencia en la productividad y competitividad de las empresas en el mercado, facilitando el análisis de los recursos, planificación y evaluar los tiempos acumulados de procesos en el sistema. La contribución de un software de simulación corresponde a una mejora de la calidad sistémica de los procesos y recursos.

**Palabras clave** – Simulador, interfaz gráfica, redes de Petri, FMS, hilos Java.

### 1. INTRODUCCIÓN

En diversos sectores productivos, la programación de actividades es de gran importancia en la incidencia de la productibilidad y competitividad de las empresas en el mercado, por lo cual es fundamental que las empresas del sector productivo se acoplen al mercado cambiante, en este contexto, las empresas se ven en la necesidad de contar con una programación de la producción efectiva. Los sectores productivos deben tener la capacidad de competir y una de las formas para lograrlo es la producción por FMS (acrónimo en herramienta de simulación que describa apropiadamente sus características y permita modelar fácilmente dichos sistemas).

En la simulación y análisis de FMS sería útil el uso de un Software de Simulación Gráfico enfocado en el usuario final permitiendo reducir el nivel de complejidad de modelamiento de los sistemas sin la necesidad que el usuario requiera de conocimientos medios o avanzados de programación e ingeniería en procesos de manufactura, para lo cual se emplea un método como las Redes de Petri (RdP) el cual es usado en varias aplicaciones para modelar y describir el comportamiento de sistemas dinámicos tratando de que su representación sea lo más cercana a la realidad.

El proyecto incursiona en el desarrollo de un Software de Simulación en base a la teoría de Redes de Petri para obtener una herramienta de análisis, diseño y modelación de FMS, los cuales se han enfrentado cada vez más a sistemas complejos debido a un entorno cambiante por lo cual aplicar nuevas teorías y métodos se ha hecho una tarea constante para mejorar su desempeño y aplicabilidad, en este caso se plantea el desarrollo de un Simulador Gráfico para racionalizar el uso de los recursos y brindar soluciones a muchos problemas complejos permitiendo una representación completa de los sistemas de manufactura.

## 2. SISTEMAS DE PRODUCCIÓN

### 2.1. Simulación

La simulación de procesos a lo largo del tiempo ha permitido resolver cuestiones complejas, una simulación es una imitación de la operación de un proceso del mundo real sobre determinado tiempo.

La simulación permite el estudio y la experimentación de las interacciones internas dentro de un sistema complejo, aparte de ser bien reconocida por varias empresas y organizaciones que la emplean hoy en día para mejorar su productividad y obtener una ventaja competitiva en medio de un mundo agresivo de constantes cambios; dentro de sus campos de aplicación tenemos: telecomunicaciones, transporte, energía, salud, economía, fabricación, distribución y logística, hasta procesos de negocios [1,2].

### 2.2. Sistemas de Manufactura Flexible

Un FMS es una celda de maquinado con tecnología de grupos altamente automatizada que consiste en un grupo de estaciones de procesamiento, interconectadas entre sí mediante un sistema automatizado de manejo y almacenamiento de materiales que a su vez son controlados por un sistema integrado de computadoras [3].

Los factores que intervienen para que el sistema de manufactura sea flexible son principalmente: la capacidad de disminuir o aumentar la cantidad de la producción, el tipo de manufactura y como esta pueda producir con una misma máquina, las diferentes operaciones que reflejan eficiencia en la elaboración de productos y el cliente a quien está enfocado todo el proceso para poder satisfacer su necesidad.

Los beneficios de un FMS son bastante amplios desde el mismo hecho que intensifica el uso de las máquinas de un proceso así como también reduce la cantidad de las mismas, entonces al disminuir la cantidad de maquinaria se reduce el espacio empleado para el sector de producción, el que puede ser aprovechado dentro de la empresa para otros enfoques. Se logran menores tiempos de entrega al evitar la existencia de los indeseados tiempos muertos o cuellos de botella [4].

### 2.3. Redes de Petri

Las redes de Petri (RdP) son una herramienta de modelado muy efectiva para la representación y el análisis de procesos concurrentes [5].

Las RdP son una clase de grafo dirigido y bipartido, conformado por dos tipos de nodos denominados lugares y transiciones. Los lugares representan, por lo general, acciones o condiciones y las transiciones representan eventos. Las plazas y las transiciones se conectan mediante arcos. En las plazas residen entidades llamadas *tokens*, que representan la información de la condición o acción asociada con la plaza.

### 2.4. Modelado de FMS con Redes de Petri

Consiste de un número  $n$  de sub redes a cíclicas que modelan las relaciones de precedencia de los  $n$  trabajos asignados al sistema de manufactura [6]. Los recursos se añaden como plazas temporizadas y no temporizadas: las primeras modelan las operaciones de los trabajos, mientras que las segundas lo hacen con las colas de almacenamiento temporal (*buffers*)<sup>1</sup> localizadas antes de las estaciones. Las transiciones modelan bien sea el inicio o el fin de una operación. La disponibilidad de las máquinas es modelada mediante plazas sin temporizar. Estas plazas, denominadas de recurso, están conectadas a las transiciones de las subredes de los trabajos y controlan que el correspondiente recurso no pueda estar utilizado por más de una máquina.

## 3. DEFINICION DE LA SOLUCIÓN

Para el desarrollo del presente proyecto, en la etapa de diseño se contemplaron los problemas de la Simulación de Sistemas de Eventos Discretos centrados en los Sistemas Flexibles de Manufactura basados

---

<sup>1</sup>*Buffers*: Memoria de almacenamiento temporal de información que permite transferir los datos entre unidades funcionales con características de transferencia diferentes.

en Redes de Petri como solución, se debe considerar que el diseño y planificación de FMS contemplan muchos componentes (máquinas, manipuladores, entradas, salidas), fabricando secuencias en escenarios complejos por lo cual obtener una herramienta de simulación gráfica que considere las tomas de decisiones para realizar todas las operaciones de forma correcta con una interfaz amigable para el usuario sería necesaria.

La razón principal para investigar este tema es encontrar una herramienta de Simulación Gráfica dirigido al usuario final con el objetivo de reducir lo máximo la dificultad del uso de la misma para poder crear cualquier tipo de FMS, por lo cual se consideran los siguientes problemas a afrontar en la creación del Simulador Gráfico:

- Desarrollo de una interfaz que considere todos los requisitos para poder diseñar una FMS.
- Ser capaz de encontrar la mejor solución basada en Redes de Petri.
- Obtener datos de la simulación para procesarlos.

Basado en los puntos anteriores se considera la necesidad de métodos para afrontar los problemas para lo cual se considera las siguientes restricciones:

- Todos los trabajos están procesados en el mismo FMS.
- Todos los materiales están a punto para procesar.
- Cada Máquina puede manejar una operación para un trabajo a la vez.
- Una vez iniciada la operación en una máquina no pueda ser detenida.
- Si dos o más trabajos necesitan ser transportados al mismo tiempo se debe dar paso a la secuencia que se encuentre más cerca de finalizar su proceso.
- Se verifica si el manipulador desplazo el producto hasta la máquina para que esta inicie su proceso
- Una forma de crear un programa factible es la consideración de usar buffer individuales o compartidos para almacenar los datos de manera temporal y mantener la secuencia de cada uno de los productos.

### 3.1. Simulación con Redes de Petri

La solución del modelo de Red de Petri a usarse en el Simulador Gráfico contempla los elementos que integran una FMS es decir máquinas y manipuladores, además de la creación de un *buffer* que ayude a la toma de decisiones.

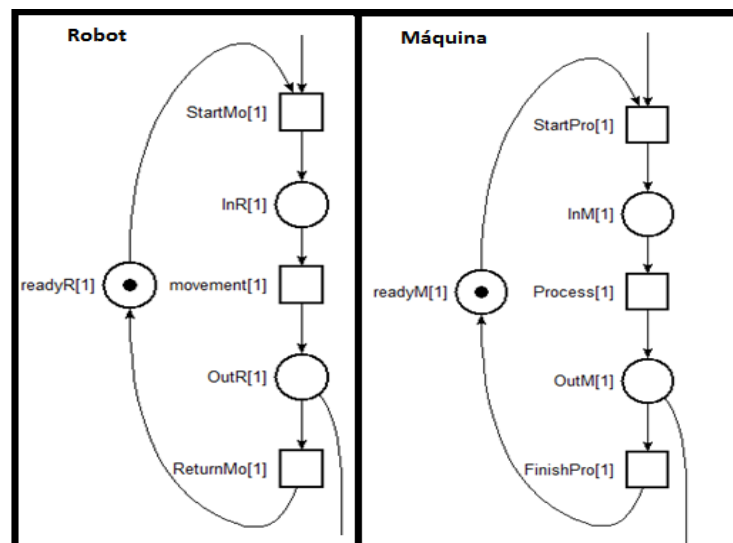


Fig. 1. Redes de Petri para Máquina y Robot.

### 3.1.1. Redes de Petri para Máquina y Robot

En la Fig. 1 se puede apreciar que la representación de una Red de Petri para una máquina es prácticamente idéntica al de un robot con diferentes nombres de variables para los lugares y las transiciones, considerando que cada lugar tiene un solo sitio que representa la capacidad de realizar un solo trabajo en cualquier tiempo, los lugares *In* y *Out* representan buffers incluidos en cada máquina o robot representando entradas o salidas de máquinas o robots, las transiciones tienen un rol importante en la simulación ya que nos permite crear los retrasos de tiempo para simular los procesos de fabricación con ayuda de la información contenida en los *tokens* de transiciones se puede programar comportamientos especiales.

También se puede apreciar en la Fig. 1 que las transiciones *StarMo* para los robots simulan el movimiento o transporte y *StarPro* simula el proceso disparando la siguiente transición hacia los lugar *OutM* y *OutR* los que representan cuando una máquina o un robot terminaron sus secuencias y están listos para el siguiente trabajo, es por esa razón que la Red de Petri cuenta con dos conexiones, la primera *readyR* o *readyM* y la segunda la que es activa por un proceso previo.

### 3.1.2. Redes de Petri para Buffer

Además de las Redes para Máquina y Robot se consideró que cada máquina y robot pueden tener un *buffer* separado o dependiendo del diseño puede tener un buffer compartido por lo cual se creó un buffer de memoria intermedia.

El diseño de una Red de Petri para un Buffer el cual sigue la lógica de diseño de la máquina y robot, con la excepción de que en el lugar *readyBuff* el cual representa la capacidad puede almacenar varios *tokens* a la vez para ser disparados dependiendo de las diferentes transiciones que se encuentren conectadas al *buffer*, y que no cuenta con una transición de transporte o proceso.

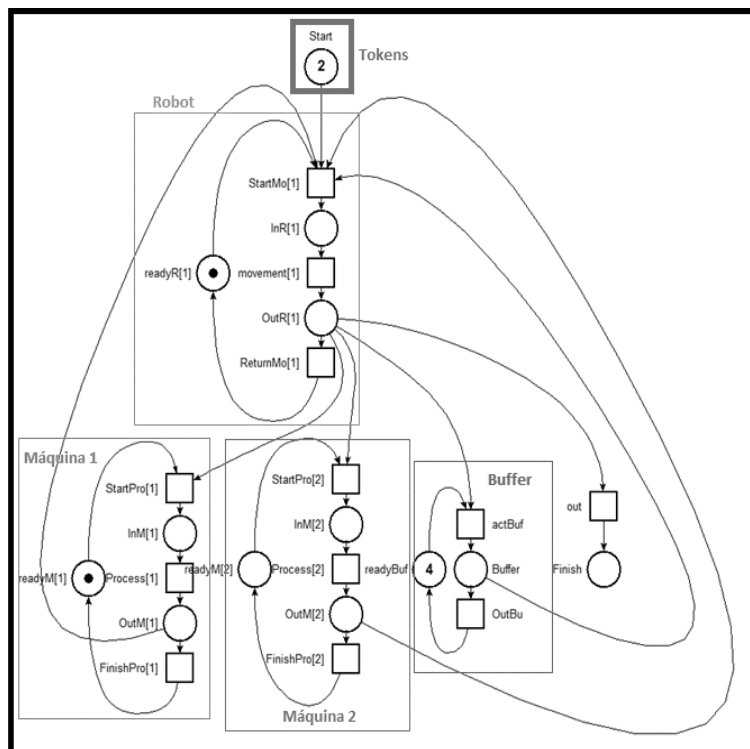


Fig. 2. Redes de Petri para Simulador Gráfico.

### 3.1.1. Redes de Petri para Simulador

Con las Redes de Petri de máquina, robot y *buffer* se crea una representación unificada para obtener una RdP para el Simulador como se observa en la Fig. 2 capaz de evaluar y simular cualquier sistema de fabricación tomando algunas consideraciones:

- La información contenida en los *tokens* puede ser modificada por cualquier transición o lugar.
- Los lugares de salidas de máquinas, robots y *buffers* se conectan a otros componentes del sistema, teniendo sincronización el caso de disparar dos o más transiciones al mismo tiempo.
- La activación de transporte para transición *StarMo* depende de un conjunto de reglas para decidir la continuación del proceso de cada producto, para el caso de tener varios procesos de transporte activos al mismo tiempo se hace uso del *buffer* que administran la activación dependiendo de la secuencia de los productos.
- El diseño de Red de Petri para el Simulador crecerá dependiendo del número de máquinas y robots existentes conectados entre sí en el diseño.

### 3.2. Interfaz Gráfica

Para el desarrollo del presente proyecto, en la etapa de diseño se tomó en cuenta los requisitos que contempla cada objeto en este caso el de máquinas, robots, salidas y entradas.

Una vez ejecutado el programa, carga una pequeña pantalla con el logo de la institución y una barra que indica el estado de carga, después de que de pasar la pantalla de carga aparece una pantalla con un menú, el cual se accede para crear un nuevo proyecto. Una vez creado un nuevo proyecto se despliega una pantalla en la cual se puede crear el diseño de un FMS con ayuda de los diferentes botones que se encuentran en la interfaz como se observa en la Fig. 3, la cual esta diferenciada por un bloque de Ingreso de Elementos, un bloque de Visualización de datos, el bloque con diferentes Listas, el Espacio o Área de Trabajo y un bloque para el Control de la Simulación.

#### 3.2.1. Bloque Ingreso de Elementos

En el bloque de Ingreso de Elementos se puede realizar el ingreso de Máquinas, Manipuladores, Entradas y Salidas, los cuales se despliegan con formas específicas en el área de trabajo y se las puedes desple-

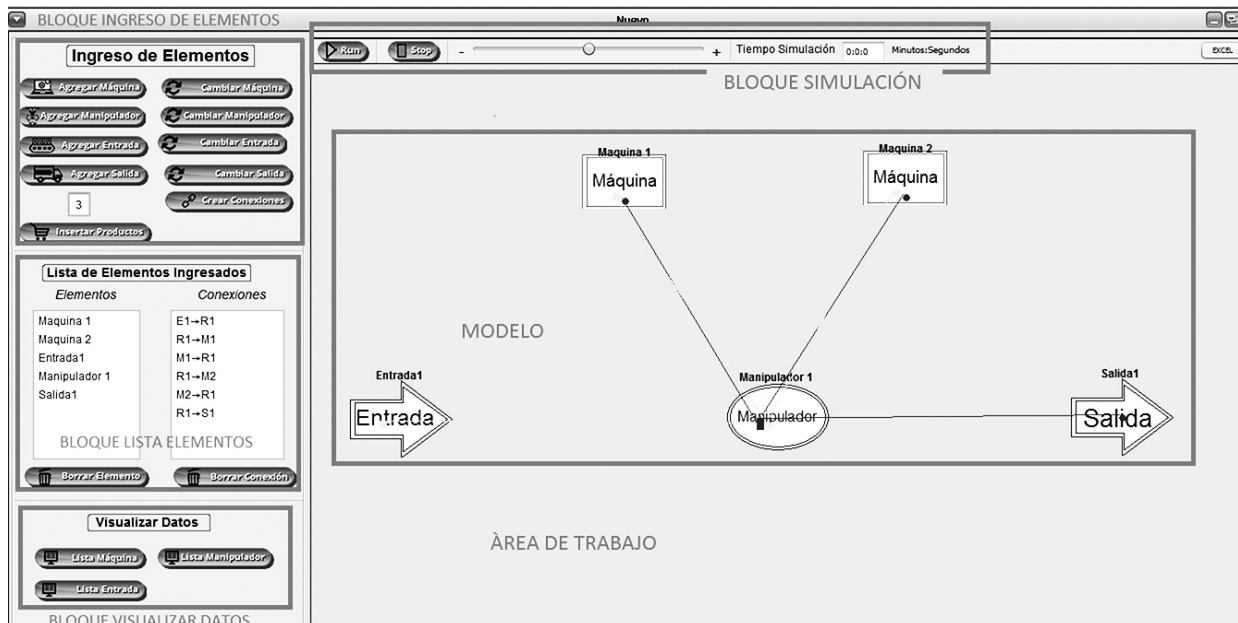


Fig. 3. Interfaz Gráfica.

gar conforme al diseño de la FMS, cada elemento cuenta con su nombre específico y numerado, en el mismo bloque también se encuentra botones con los cuales se puede cambiar el color de cada uno de los elementos para que exista una mejor diferenciación y un diseño más llamativo para el usuario.

Una vez ingresados los elementos necesarios para el diseño de un FMS se los conecta con el botón Conexiones que se encuentra en el bloque de Ingreso de Elementos dependiendo del diseño elegido con algunas restricciones que el programa contiene como:

- El elemento Entrada no puede ser conectado como Registro de Entrada.
- El elemento Salida no puede ser conectado como Registro de Salida.
- No se puede crear dos veces la misma conexión.

Además cada conexión se diferencia por dos colores evitando la confusión en el sobre dibujado de las mismas, además en cada extremo se dibuja un cuadrado que representa registro de salida y un círculo en el registro de entrada.

El número de tipos de producto se los puede realizar también en el bloque de Ingreso de Elementos con ayuda del botón Ingresar Productos considerando que el tipo de número debe ser tipo entero.

### 3.2.2. Bloque Lista de Elementos

Los elementos y conexiones ingresados en el área de trabajo también se los puedes observar en la Listas desplegable de Elementos y Conexiones, en el orden en el que fueron creados, esta lista nos ayuda para eliminar o modificar los elementos en el área de trabajo con ayuda de los diferentes botones que se encuentran en la interfaz.

Las Listas de Conexiones se crear a partir de un código formado por la primera letra del elemento y su número respectivo dependiendo del tipo de elemento como:

- **Entrada:** Código (E) + Número Identificador (1...n)
- **Máquina:** Código (M) + Número Identificador (1...n)
- **Manipulador:** Código (R) + Número Identificador (1...n)
- **Salida:** Código (S) + Número Identificador (1...n)

### 3.2.3. Bloque Área de Trabajo

En el Área de Trabajo una vez dispuesto el diseño del FMS se puede configurar cada uno de los elementos dando doble clic en el respectivo elemento el cual abrirá una nueva ventana con las propiedades a modificarse el cual guardara la información en las bases de datos.

Las Propiedades a modificarse de Máquina contemplan el número de piezas a procesarse, el tiempo de proceso, el tipo de producto y por defecto el nombre de la máquina. En las Propiedades de Manipulador solo se configura el tiempo de transporte de cada una de las conexiones realizadas previamente y por defecto las salidas-entradas de las conexiones en forma de código con las casillas desactivadas para la modificación.

En las propiedades de entrada se configura en una primera ventana el número de secuencias necesario para el diseño de FMS basado en el número de conexiones existentes. Una vez configurado este número se abre una ventana nueva donde se inserta los datos del total de productos de cada tipo y se crea las secuencias de operaciones para cada producto basado en las conexiones realizadas previamente.

### 3.2.4. Bloque Visualización de Datos

Se puede realizar la visualización de datos de las listas de Máquina, Manipulador y Entrada en cada una de estas listas se observa los datos guardados de los requisitos que cada uno de estos elementos necesitan para configurar el FMS. Al ingresar en la Lista de Máquina se despliega una ventana con los datos respectivos de código de máquina, número de producto, capacidad de la máquina y tiempo de proceso, en esta ventana se puede modificar y eliminar los campos si es necesario.

En la Lista Manipulador se puede observar directamente los datos guardados de las conexiones existentes y poder modificar los campos de tiempo de transporte. La Lista Entrada con la opción de poder modificar directamente las Propiedades de Entrada.

### 3.2.5. Bloque de Simulación

Este último bloque contiene los botones para iniciar, detener la simulación, control de velocidad de la misma, el tiempo total de la simulación realizada y un botón para exportar los datos obtenidos al simular el FMS en formato .xls.

## 4. DEFINICION DE LA SOLUCIÓN

### 4.1.1. Descripción del Software

El Simulador Gráfico basado en Redes de Petri fue diseñado en lenguaje de programación libre Java con el objetivo de reducir el nivel de complejidad de moldeamiento de los sistemas de manufactura flexibles sin la necesidad que el usuario final requiera de conocimientos avanzados y sea amigable en su funcionamiento.

En la Fig. 4 se observa la descripción de los pasos o actividades que el Usuario Final debe llevar a cabo para generar una simulación de un modelo FMS en el Simulador Gráfico, según el Diagrama de Caso de Uso se realizó una distribución de 18 clases con 10 interfaz relacionadas entre sí, donde el usuario va a poder manipular la aplicación de forma sencilla, estos archivos constan de las clases y métodos adicionales necesarios para manejar funcionalidades comunes, además se buscó aplicar la mejor forma para la reutilización de código, diseñando métodos que permitan tener centralizadas ciertas funciones para el mejor uso del sistema.

En el desarrollo del software para el Simulador Gráfico se debe considerar el estudio de Hilos o *Thread*, ya que se conecta directamente con el lenguaje Java y se aplicó como herramienta para constitución y creación de las Redes de Petri. Un hilo es un proceso en un programa ejecutándose dentro de su propio espacio de dirección, Java es sistema multiproceso, esto significa que soporta varios procesos corriendo a

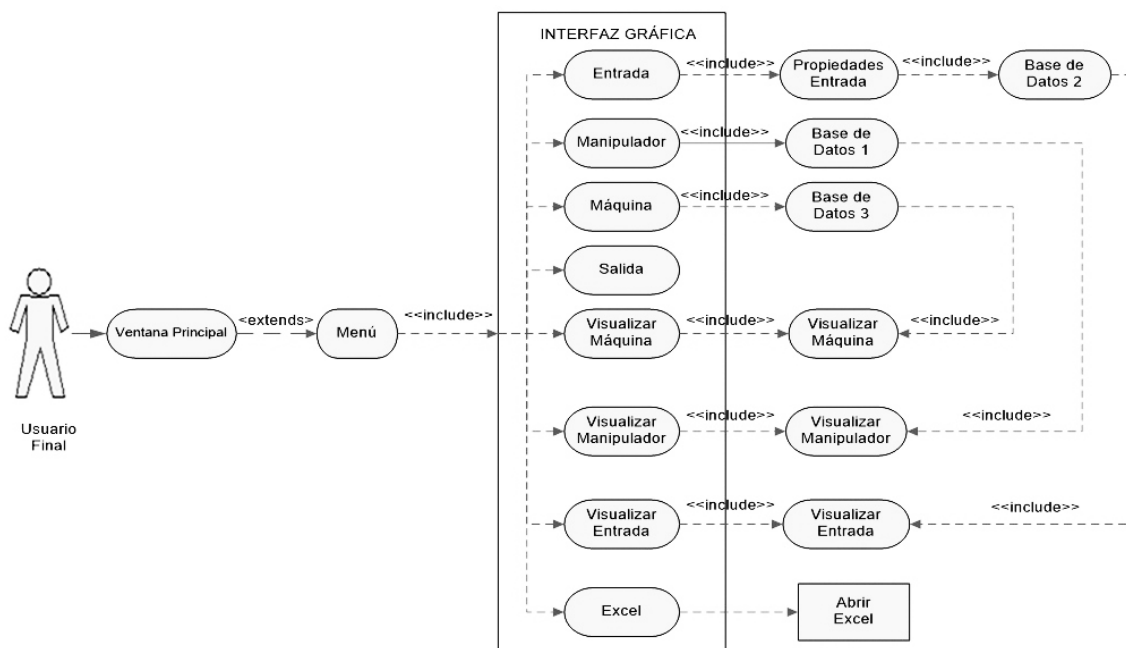


Fig. 4. Diagrama de Uso Simulador Gráfico.

la vez dentro de sus propios espacios de direcciones, un hilo es una secuencia de códigos en ejecución dentro del contexto de un proceso, los hilos no pueden ejecutarse solos, requieren de la supervisión de un proceso padre para correr, los hilos a menudo son conocidos o llamados procesos ligeros muy similar a un proceso con la diferencia de que un hilo siempre corre dentro del contexto de otro programa. Los procesos mantienen su propio espacio de direcciones y entorno de operaciones, los hilos dependen de un programa padre en lo que se refiere a recursos de ejecución [7].

#### 4.1.2. Diagrama de Flujo

En el Diagrama de Flujo de la Fig. 5, se describe el funcionamiento interno del Simulador Gráfico basado en Redes de Petri e Hilos de Java, en el cual que con ayuda de la interfaz gráfica se puede ingresar los datos en las diferentes bases de datos considerando que antes de iniciar la simulación el programa verifica si estos datos fueron ingresados correctamente como se observa en el diagrama, una vez verificados los datos el programa calcula la suma total de los productos de cada lote y crea hilos con el nombre (Nuevo Hilo) más un ID que va desde el primer producto hasta  $n$  o el último producto ( $i \dots n$ ), estos hilos contienen en su interior *tokens* o información de cada producto que para cada uno sería:

- Número de Lote

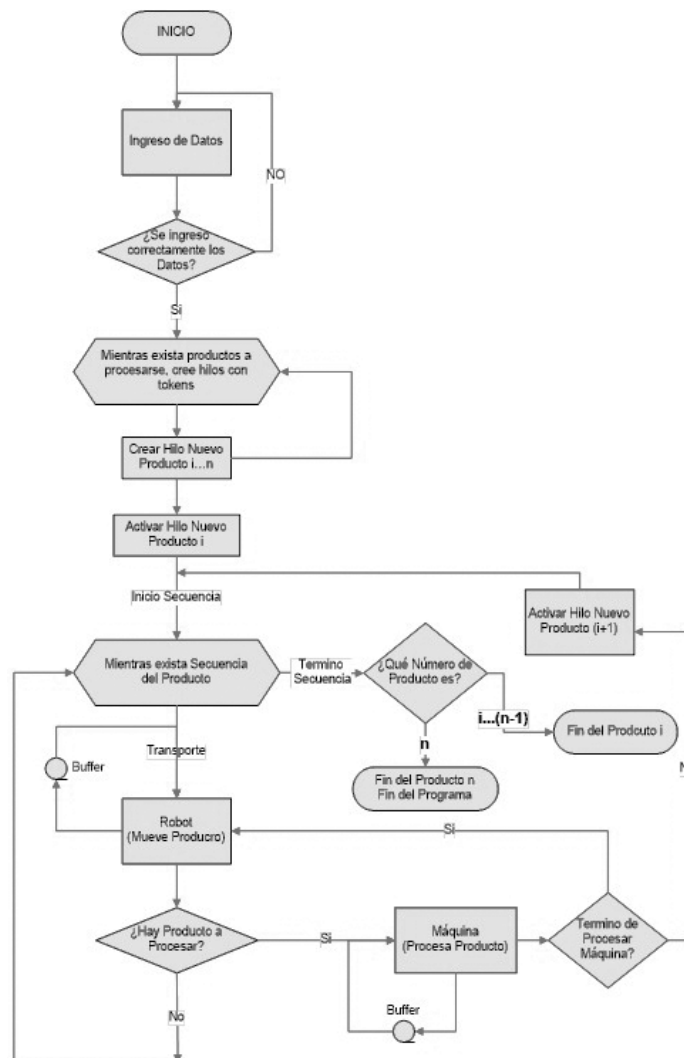


Fig. 5. Diagrama de Flujo para Simulador Gráfico.



- Número de Producto
- Secuencia del Lote
- Tiempos de Producto para cada Máquina y Robot

Cada una de estas secuencias tienen el propósito de activar el transporte del robot y verificar si existe o no producto a procesarse como se observa en diagrama, se debe acotar que los *buffers* del diagrama son la representación del *Buffer* para las Redes de Petri así como también las de Máquina y Robot, siguiendo con el flujo del diagrama si la máquina se encuentra ocupada nos informa que el robot está listo para seguir procesando por lo cual da una señal para activar un Hilo Nuevo con siguiente Producto  $i+1$ , así sucesivamente hasta llegar al Producto  $n$ , y si la maquina termino su proceso activa otra señal para que el producto asignado siga con su secuencia. Una vez terminada la secuencia de cada producto el programa compara que ID de producto finalizo para determinar si es el fin para un producto terminado ( $i \dots n-1$ ) o el fin del último producto ( $n$ ) el cual finaliza la simulación.

## 5. DEFINICION DE LA SOLUCIÓN

En la prueba de resultados del Sistema del Simulador Gráfico se utilizó un modelo para resolver un Caso de Estudio en el que se debe programar cinco pedidos que cubre cinco tipos de partes diferentes para ser producidos en un Sistema de Manufactura Flexible, por lo cual el diseño del modelo está compuesto por dos tipos de máquinas distintas, de tipo A y tipo B. Las máquinas son preparadas una vez al día y deben estar disponibles por una producción continua. El objetivo es programar y calcular el tiempo requerido para cada tipo lote que se producirán en los siguientes días basados en los datos que se resumen en la tabla 1.

En la Fig. 6 se comprueba que los datos ingresados en el Caso de Estudio fueron procesados correctamente por el Simulador Gráfico, cumpliendo con el pedido de cada uno de los lotes con un total de 54 productos de diferentes tipos (A, B, C, D y E) en un tiempo total de 5 horas 44 minutos y 20 segundos.

Con ayuda del Diagrama de Gantt del Simulador se pudo obtener los tiempos de inicio y fin de cada uno de los 5 lotes producidos como se observa en la tabla 2, también se calculó la utilización de cada máquina que intervino en el proceso de producción de cada lote como se puede observar a continuación en la tabla 3.

Tabla 1. Datos del Problema.

Tiempo de Procesado por Unidad (min)			
Tipo de Parte	Tamaño de Orden	Máquina Tipo A	Máquina Tipo B
<b>A</b>	5	3	1
<b>B</b>	10	12	0
<b>C</b>	25	7	4
<b>D</b>	10	2	1
<b>E</b>	4	3	2

Tabla 2. Tiempos de Producción por Lote.

Producción por Lote				
Número de Lote	Número de Productos	Tiempo de Inicio	Tiempo Final	Makespan
<b>Lote 1</b>	5	0:00:00	0:15:59	0:15:59
<b>Lote 2</b>	10	0:14:59	2:15:00	2:00:00
<b>Lote 3</b>	25	2:15:00	5:15:00	3:00:00
<b>Lote 4</b>	10	5:10:00	5:31:01	0:21:00
<b>Lote 5</b>	4	5:30:00	5:44:02	0:14:00



- [3] Jaramillo, C., “Modelación de Células de Manufactura Flexible mediante Redes de Petri y Autómatas Celulares”. Tutor: Medina Joselito. Tesis de Maestría. Centro de Investigación Avanzada de Ingeniería Industrial, Universidad Autónoma del Estado de Hidalgo, España (2012)
- [4] Grove, M., *Automation, Production System and Computer Integrated Manufacturing*, E.E.U.U, New Jersey: Prentice Hall Press Upper Saddle River (2002)
- [5] Sethi, A.K., Sethi, S.P., “Flexibility in Manufacturing: A survey”, *International Journal of Flexible Manufacturing Systems*, 2, 289 (1990)
- [6] Caballero, J.P., Mejía, G., “Redes de Petri y Algoritmos Genético, una Propuesta para la Programación de Sistemas de Manufactura Flexible”, Colombia, Bogotá: Universidad del Bogotá (2006)
- [7] Inmaculada, G., Sánchez, A., Hernández, D., “Java Threads (Hilos en Java)”. Tutor: García Francisco. Tesis de Pregrado. Departamento de Informática y Automática, Universidad de Salamanca, España (2002)

### **DEVELOPMENT OF A SIMULATOR OF MANUFACTURING SYSTEMS WITH GRAPHIC INTERFACE BASED ON PETRI NETWORKS**

**Abstract** – This research develops ideas in the application of Discrete Even Systems (DES) simulation software for the optimization of Flexible Manufacturing Systems (FMS), the DES have intent to identify systems in which events change in a discrete manner one after another, in specific instants of time. Although seemingly simple, DES can model many phenomena one application is the productive process of a company for which the proposed development of a Simulator with a graphical interface programmed in Java, supported by Petri Nets for represent the dynamic behavior of systems and use it as a tool for influencing productivity and competitiveness of manufacturing companies, facilitating analysis of resources, planning and evaluate different metrics related with production times in the system. The contribution of simulation software corresponds to an improvement for systemic quality that allows companies to obtain products at a reasonable cost, with increased flexibility and more over optimizing resources.

**Keywords** – Simulator, Graphical Interface, Petri Nets, FMS, Java Threads.