

DISEÑO DE UN SISTEMA AUTOMATIZADO PARA EL MONITOREO CONTINUO EN TIEMPO REAL DE LOS COSTOS DE ENSAMBLAJE DE CIRCUITOS IMPRESOS

Por
Enrique Saavedra Sada

Proyecto sometido en cumplimiento parcial de los requisitos para el grado de

MAESTRO EN INGENIERIA

en

SISTEMAS GERENCIALES

**UNIVERSIDAD DE PUERTO RICO
RECINTO UNIVERSITARIO DE MAYAGÜEZ**

2011

Aprobado por:

Agustín Rullán, Ph.D.
Presidente, Comité Graduado

Fecha

William Hernández, Ph.D.
Miembro del Comité Graduado

Fecha

Pedro Resto, Ph.D.
Miembro del Comité Graduado

Fecha

Mario Padrón, Ph.D.
Miembro del Comité Graduado

Fecha

Viviana Cesaní, Ph.D.
Director del Departamento

Fecha

Silvestre Colón, Prof.
Representante de la Escuela Graduada

Fecha

"La incompetencia comienza cuando tratamos de resolver los problemas de hoy utilizando herramientas de ayer".

- Albert Einstein -

RESUMEN

Se creó un prototipo de un sistema capaz de recopilar y analizar información de costos de ensamblaje en tiempo real. Se usó una línea de ensamblaje de paneles de circuitos impresos disponible en el Departamento de Ingeniería Industrial de la Universidad de Puerto Rico en Mayagüez conocida como la Fábrica Modelo. Para crear dicho prototipo se analizaron las estaciones de la línea de ensamblaje de circuitos impresos y se determinó cómo se podía obtener la información relevante de cada una de ellas para calcular los costos, así como el tiempo de proceso, la materia prima que se consumía y los errores que sucedían en las estaciones. Una vez que se obtuvo esa información, se hicieron los cálculos pertinentes para obtener los costos reales por estación. Simultáneamente, se hicieron cálculos por estación utilizando ecuaciones conocidas para obtener los costos teóricos, y se calcularon los costos de producción en el tiempo real. Se creó una gráfica y un reporte en los que se compararon los costos reales con los costos teóricos. Dicha gráfica fue de gran utilidad, porque se pudo apreciar esquemáticamente el comportamiento de los costos de las unidades producidas. Este proyecto se puede aplicar en diversas compañías automatizadas y/o semi-automatizadas, y puede ser de gran utilidad porque proporciona a la gerencia los costos de producción en tiempo real y conociendo esa información, se pueden tomar decisiones oportunamente.

ABSTRACT

We created a prototype of a system capable of collecting and analyzing real time assembly cost data information. We selected a printed circuit boards (PCB) assembly line available at the Department of Industrial Engineering, University of Puerto Rico at Mayagüez, known as “Fábrica Modelo”. To create this prototype the PCB line was analyzed to determine how we could obtain data information such as relevant cost and processing times, raw materials used, and the historical number of errors for each station. Once the information was obtained, we calculated the real costs. Simultaneously, some calculations were made for each station using well-known equations to obtain the real time theoretical and production costs. A graphical analysis was obtained to compare real time and theoretical costs. The graph is very useful to track the units produced costs behavior. This project can be applied to many automation and/or semi-automation companies, and it can be very useful because it can provide production cost data in real time and with that information management can make informed decisions.

AGRADECIMIENTOS

Durante mis estudios graduados en la Universidad de Puerto Rico, muchas personas e instituciones cooperaron directa e indirectamente en mi investigación. Sin esa ayuda, no me hubiera sido posible terminar este proyecto. Quisiera dedicarles este reconocimiento. En primer lugar, le quiero agradecer a Dios por haberme dado la oportunidad de hacer esta maestría.

Agradezco de forma especial, al Dr. Agustín Rullán, por su ayuda, paciencia y guía durante esta etapa de mis estudios. También agradezco a los demás miembros de mi comité: Dr. Mario Padrón, Dr. William Hernández y al Dr. Pedro Resto por su apoyo y consejos durante estos años de estudios graduados. También quiero agradecer al Dr. Mauricio Cabrera por sus consejos y su apoyo siempre que lo necesité desde que comencé estos estudios graduados.

También agradezco al Ing. Freddy Jusino, y al Sr. Luis Jiménez, dos amigos que sin su ayuda no hubiera logrado los objetivos que me propuse en este proyecto. Agradezco muy especialmente al Ingeniero Ramón Cancel, supervisor de la Fábrica Modelo, que gracias a su ayuda y a su disponibilidad, siempre me facilitó el trabajo, y a los demás estudiantes que trabajaron en la Fábrica durante mi proyecto, y siempre estaban disponibles para cooperar conmigo en lo que hiciera falta. Al Sr. Israel Tirado, que fue de gran ayuda en todo lo que se refería a la red de computadoras del Departamento.

Agradezco también a todos mis compañeros de estudios graduados y a todos los miembros del Departamento de Ingeniería Industrial por su valiosa ayuda durante estos años. Finalmente, agradezco a todos los que de alguna u otra forma contribuyeron a que lograra este objetivo.

TABLA DE CONTENIDO

Lista de Figuras	viii
Lista de Gráficas.....	ix
Lista de Ecuaciones	x
Lista de Acrónimos	xiii
Lista de Apéndices	xiv
1. Introducción	1
2. Trabajos Previos.....	3
2.1 Proyectos Similares de Monitoreo en Tiempo Real	3
2.2 Tecnología Usada en el Monitoreo de Procesos en Tiempo Real	4
2.3 Modelos para el Monitoreo de Costos en Tiempo Real	6
3. La Fábrica Modelo como plataforma para la creación de un Prototipo.....	9
4. Modelo de Costos	14
4.1 Cálculos de los Costos en el Tiempo Real	14
4.2 Cálculos de los costos teóricos	25
5. Diseño e implementación del sistema prototipo	26
5.1 Conexiones y Aparatos Físicos Utilizados para la Captura de Datos.....	26
5.1.1 Estaciones en las que se Puede Acceder al “ <i>Firmware</i> ”	26
5.1.2 Estaciones en las que No se Tiene Acceso al “ <i>Firmware</i> ”	27
5.2 Programación Llevada a cabo para Procesar la Información de Costos.....	28
5.2.1 Módulo de Adquisición de Datos en Tiempo Real de las Estaciones con PLC	28
5.3 Interfaz con el Usuario	31

5.4 Archivo “Stacomb.dat”	31
6. Resultados y trabajos futuros	33
7. Referencias.....	35
8. Apéndices.....	37

LISTA DE FIGURAS

Figura 1 Diagrama del Proceso de Ensamblaje.....	10
Figura 2 Despliegue de Facilidades de la Fábrica Modelo	10
Figura 3 Estación 1: Serigrafía.....	11
Figura 4 Estación 2: Máquina de Ensamblaje I.....	11
Figura 5 Componentes que ensambla la máquina de Ensamblaje I.....	11
Figura 6 Componentes que ensambla la máquina de Ensamblaje II.....	12
Figura 7 Estación 3: Máquina de Ensamblaje II	12
Figura 8 Estación 4: Inspección	13
Figura 9 Estación 5: Horno	13
Figura 10 Esquema del cableado de las estaciones 2 y 3.....	26
Figura 11 Esquema del cableado de las estaciones 1, 4 y 5.....	27
Figura 12 Conexiones del PLC a la PC por el OPC.....	28
Figura 13 Diagrama del Programa	29
Figura 14 Diagrama de la programación para obtener datos del Archivo Eventdata.txt	30
Figura 15 Gráfica final	34
Figura 16 Reporte final de Costos.....	34
Figura 17 Reporte final	96

LISTA DE GRÁFICAS

Gráfica 1 Gráfica final.....	96
------------------------------	----

LISTA DE ECUACIONES

(01) Costo de Manejo de Materiales	17
(02) Costo de Mano de Obra Ingeniería.....	17
(03) Costo de Preparación de la Est.1	17
(04) Costo del Plomo por Panel	18
(05) Costo del Panel.....	18
(06) Costo Energía Bombillas Est.1	18
(07) Costo Energía Aire Acon. Est.1.....	18
(08) Costo Energía Est.1	18
(09) Costo Energía Compresor Est.1.....	18
(10) Costo Energía Computadoras Est.1	18
(11) Costo Total Energía Est.1	18
(12) Costo de Otros Est.1	19
(13) Costo de Preparación Est.2.....	19
(14) Costo de M. de O. Cambio Rollos Est.2.....	19
(15) Costo Energía Bombillas Est.2.....	20
(16) Costo Energía Aire Acon. Est.2.....	20
(17) Costo Energía Est.2	20
(18) Costo Energía Compresor Est.2.....	20
(19) Costo Energía Computadoras Est.2	20
(20) Costo Total Energía Est.2.....	20
(21) Costo de Otros Est.2	20
(22) Costo de Preparación de la Est.3	20
(23) Costo de M.de O. Cambio Rollos Est.3.....	20
(24) Costo Energía Bombillas Est.3	21

(25) Costo Energía Aire Acond. Est.3.....	21
(26) Costo Energía Est.3	21
(27) Costo Energía Compresor Est.3.....	21
(28) Costo Energía Computadoras Est.3	21
(29) Costo Total Energía Est.3	21
(30) Costo de Otros Est.3	21
(31) Costo de Preparación Est.5	22
(32) Costo del Nitrógeno por panel.....	22
(33) Costo Energía Bombillas Est.5.....	22
(34) Costo Energía Aire Acond. Est.5.....	22
(35) Costo Energía Est.5	22
(36) Costo Energía Computadoras Est.5	23
(37) Costo Total Energía Est.5.....	23
(38) Costo de Otros Est.5	23
(39) Costo de Mano de Obra Inspección.....	23
(40) Costo de Energía Bombillas Inspección.....	23
(41) Costo de Energía Aire Acond. Inspección.....	23
(42) Costo de Energía Computadoras Inspección.....	23
(43) Costo Total de Energía Inspección.....	23
(44) Suma de costos Estación 1.....	24
(45) Suma de costos Estación 2.....	24
(46) Suma de costos Estación 3.....	24
(47) Suma de costos Estación 4.....	24
(48) Suma de costos Estación 5.....	24

(49) Suma de los costos totales de las estaciones.....	24
(50) Suma de los costos de setup de las estaciones.....	24
(51) Suma de los costos de Otros de las estaciones	25
(52) Suma de costos por panel	25

LISTA DE ACRÓNIMOS

ABC	Activity-Based Cost
C/C	Fuji Communication Center
DDE	Dynamic Data Exchange
DOS	Disk Operating System
EBI	Electro Biology Inc.
OEE	Overall Equipment Effectiveness
OLE	Object Linking Embedding
OPC	OLE for Process Control
PCB	Printed Circuit Boards
PLC	Programmable Logic Controller
SMT	Surface Mount Technology
UPRM	Universidad de Puerto Rico, Mayagüez
VB6	Visual Basic 6
VBA	Visual Basic for Applications
VB.NET	Visual Basic.NET
WIP	Work In Process

Nota: En este proyecto se utilizó la palabra “*firmware*”, pues es un término que en español requiere muchas palabras para definirlo: Es un programa que se imprimió dentro de los circuitos electrónicos del ordenador o en su memoria ROM y que no puede ser modificada por el usuario.

LISTA DE APÉNDICES

Apéndice 1 Activar/Desactivar el "Remote Desktop" en Windows 7	37
Apéndice 2 Configurar la computadora con Windows 7 para utilizar el OPC DataHub.....	39
Apéndice 3 Módulo "Datoscostos.vba". Interface con el usuario	44
Apéndice 4 Módulo "Todas las estaciones"	50
Apéndice 5 Módulos "Station 2.exe y Station 3.exe"	51
Apéndice 6 Módulo "Control.exe"	74
Apéndice 7 Módulo de VBA® para llenar el archivo "stacomb.dat" de la primera estación	78
Apéndice 8 Módulo de VB6® para llenar el archivo "stacomb.dat" de la segunda estación	82
Apéndice 9 Módulo de VB6® para llenar el archivo "stacomb.dat" de la tercera estación	84
Apéndice 10 Módulo de VBA® para llenar el archivo "stacomb.dat" de la cuarta estación.....	86
Apéndice 11 Módulo de VBA® para llenar el archivo "stacomb.dat" de la quinta estación	90
Apéndice 12 Gráfica y Reporte final.....	96

1. INTRODUCCIÓN

El ambiente en la industria de la manufactura es uno que requiere de mucha agilidad. Es necesario obtener información en tiempo real para poder reaccionar rápidamente a los cambios que surgen y atender los problemas a la brevedad posible. La tecnología disponible hoy día en la industria de la manufactura está lo suficientemente avanzada como para implantar sistemas que permiten obtener información de todo tipo en tiempo real. Algunos ejemplos de esta tecnología lo son; sensores, redes de computadoras, estándares de comunicación de datos, gran variedad de controladores lógicos programables, en fin, una gran diversidad de aparatos y equipos para obtener y procesar información en tiempo real. Según Frontini y Kennedy [1] "Con la aplicación de esta tecnología, se puede llegar a ver lo que va a ser la próxima generación de máquinas en la manufactura".

Muchas empresas de manufactura han comenzado a recopilar y usar información en tiempo real para darles retroalimentación a los operadores sobre su ejecutoria hasta el momento [2]. Se pueden ver pantallas de computadora, por ejemplo, en estaciones de ensamblaje con este tipo de información. Sin embargo, no se suele recopilar información relacionada a costos en tiempo real. Esta información podría ser de gran utilidad para tomar decisiones y contestar preguntas de importancia inmediata como las siguientes. ¿Están costando los productos más o menos de lo esperado? ¿Se recuperará la inversión, los costos directos y los costos indirectos? ¿Requiere algún proceso alguna atención especial debido a alguna desviación de lo esperado? ¿A qué le debe prestar atención la gerencia para mejorar los procesos? ¿Se está aprovechando toda la materia prima?

En este proyecto se creó un prototipo de un sistema capaz de recopilar y analizar información de costos de ensamblaje en tiempo real. Se usó una línea de ensamblaje de paneles de circuitos impresos disponible en el Departamento de Ingeniería Industrial de la Universidad de Puerto Rico en Mayagüez conocida como la Fábrica Modelo. Para crear dicho prototipo se analizaron las estaciones de la línea de ensamblaje de circuitos impresos y se determinó cómo se podía obtener la información relevante de cada una de ellas para calcular los costos, así como el tiempo de proceso, la materia prima que se consumía y los errores que sucedían en las estaciones. Una vez que se obtuvo esa información, se hicieron los cálculos pertinentes para obtener los costos reales por estación. Simultáneamente, se hicieron unos cálculos por estación utilizando unas ecuaciones conocidas para obtener los costos teóricos. Una vez obtenida la información necesaria, se calcularon

los costos de producción en el tiempo real, lo que permitió que se pudieran contestar algunas de las interrogantes arriba mencionadas.

En cada una de las estaciones, se fueron sumando los costos que contribuyen al costo final de producción. En algunas de las estaciones, únicamente se obtuvo el tiempo de producción. En otras estaciones, por la naturaleza de la estación, se obtuvo más información, y se pudo saber si durante el ensamble hubo algún tipo de problemas, como por ejemplo errores y/o componentes desperdiciados, algo que también influye en el costo final. Se creó una gráfica en la que se comparó, en tiempo real, los costos reales con los costos teóricos. Dicha gráfica fue de gran utilidad, porque se pudo apreciar esquemáticamente el comportamiento de los costos de las unidades producidas.

En la segunda sección de este documento, se analizaron algunos estudios relevantes y trabajos previos relacionados al tema del proyecto. En la tercera sección, se habló del lugar donde se diseñó e implantó este proyecto: en la Fábrica Modelo de la Universidad de Puerto Rico. En la cuarta sección, se describieron los modelos de costos que se utilizaron, al igual que las ecuaciones. En la quinta sección se describió detalladamente la tecnología, tanto de *"hardware"* como de *"software"* que se utilizó y se explicaron detalladamente cada uno de los programas que se hicieron. En la última sección, se recogieron todos los resultados que se obtuvieron y algunos trabajos futuros que se podrían hacer para ampliar este proyecto.

2. TRABAJOS PREVIOS

En este apartado se van a mencionar algunos de los trabajos que se han realizado y que tienen relación con este proyecto en los temas de monitoreo de un proceso automatizado en tiempo real; de las principales tecnologías para configurar este sistema, tanto de “*hardware*” como de “*software*”; de la línea de manufactura donde se aplicó este proyecto y del sistema de costos utilizado.

2.1 PROYECTOS SIMILARES DE MONITOREO EN TIEMPO REAL

En [3] describen un sistema donde se monitorea, en tiempo real, el llenado de botellas de agua pura para beber. En dicho sistema se utilizó un Control de Supervisión y Adquisición de Datos¹, es un sistema basado en computadoras que permite supervisar y controlar variables de proceso a distancia, proporcionando comunicación con los equipos de la fábrica y controlando el proceso de forma automática por medio de un software especializado.

En [4] presentan un modelo de supervisión para el control “*online*” de centrales de enfriamiento de agua para mejorar su eficiencia energética y así utilizar la menor cantidad de energía para proporcionar la adecuada refrigeración de los edificios, teniendo en cuenta las características y las interacciones de los sistemas centrales de refrigeración.

La compañía Azeotech, es una compañía líder en software para medición y automatización [5], ofrece un software que proporciona un control de supervisión y adquisición de datos SCADA y la interfaz hombre-máquina HMI² para conocer determinada información en tiempo real. Han instalado software en distintos países en el mundo, y en gran diversidad de áreas, como en la industria aeroespacial, en la agricultura, en la auto locomoción, en la distribución de energía, en la administración de edificios, en la manufactura, en la minería, en varias aplicaciones militares, en investigaciones científicas, tecnológicas y educativas entre otras. Este software es una herramienta poderosa y flexible.

¹ SCADA por las siglas de *Supervisory Control and Data Acquisition* en inglés.

² Interface Humano-máquina. HMI por sus siglas en inglés (*Human Machine Interface*): una interfaz gráfica que permite a una persona interactuar con un sistema de control.

2.2 TECNOLOGÍA USADA EN EL MONITOREO DE PROCESOS EN TIEMPO REAL

Entre las principales tecnologías que se han utilizado en sistemas de monitoreo de procesos en tiempo real se pueden mencionar, computadoras personales, controladores lógicos programables, sensores y OPC, entre otros. A continuación se definirán cada una de estas tecnologías y cómo fueron usadas.

Un sensor es un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que se pueda cuantificar y/o manipular. Normalmente se usa para detectar alguna característica en el proceso. Se puede indicar la presencia o ausencia de un producto en una estación, proximidad de algún objeto, temperatura de un horno, color, etc. Los sensores vienen a ser uno de los elementos más importantes de un sistema automatizado, dado que proveen al controlador parte del estado del proceso necesario para la toma de decisiones.

Un Controlador Lógico Programable (PLC por sus siglas en inglés) “es un microprocesador programable inventado para reemplazar los circuitos secuenciales de relés utilizados en el control de máquinas. El PLC trabaja revisando el estado de sus entradas, y de acuerdo a esos estados, manipula el estado de sus salidas encendiéndolas o apagándolas” [6]. El PLC es para trabajar con el *hardware*.

Una Computadora Personal, es un aparato capaz de realizar una serie de operaciones aritméticas o lógicas, sin necesidad de conectarse a otra computadora más poderosa, diseñado para ser manejado por una persona [7]. En este proyecto, se utilizó una computadora personal (PC) por ser más apropiada para el manejo de información, para hacer los cálculos y para hacer la gráfica, porque, al haber software especializado para la PC, permite manejar la información más fácilmente. La computadora que se utilizó, es una computadora marca Dell, con sistema operativo Windows XP Professional, con 4 puertos USB, uno ethernet y uno serial (RS-232). En esa computadora se almacenó la información de los tiempos de proceso de las diferentes estaciones, luego se calcularon los costos teóricos y los reales, se hicieron la gráfica y el reporte final comparando esos costos.

Todas estas cosas que se hicieron en la PC, son imposibles hacerlas en un PLC. La PC tiene instalado el programa Office de Microsoft[®], concretamente se utilizó el programa MSeExcel[®].

Para que una computadora personal pueda recibir información de un PLC, es conveniente utilizar un *software* intermedio que implante un estándar de comunicación entre máquinas conocido como OPC. El objetivo del estándar OPC es que máquinas de distintos fabricantes se puedan comunicar entre ellas con cierta facilidad. Un OPC es un *software* que permite comunicar equipos industriales con programas Windows y/o programas no Windows, el OPC recibe los datos y los transmite, tiene una arquitectura: servidor/cliente.

El **OPC servidor** es un programa que convierte la señal de entrada del PLC al protocolo de comunicación que puede ser entendido por un OPC Cliente. El **OPC cliente**, es un programa que necesita conectarse a un *hardware*, como un HMI. El OPC cliente, utiliza el OPC servidor para obtener o enviar información al equipo industrial. Es un “estándar abierto”, lo que significa menores costos para los fabricantes y más opciones para los usuarios. Los fabricantes de equipos industriales sólo necesitan dar un único servidor OPC. Los proveedores de *software* deben incluir la capacidad de que sus productos se puedan convertir en **cliente OPC** y al instante se hacen compatibles con miles de equipos industriales. Los usuarios pueden elegir cualquier *software* como **cliente OPC** con la seguridad de que van a comunicarse sin problemas con el OPC servidor habilitado para su equipo. Existe un OpenOPC para Python³ que se puede obtener en la siguiente dirección de internet: <http://openopc.sourceforge.net>.

En varios de los artículos sobre proyectos con OPC [6,8,9] se resalta la ventaja de la flexibilidad de este programa de computadora para obtener datos de distintos equipos y de distintos fabricantes. Esta flexibilidad hace que los costos y los tiempos de implementación estén muy por debajo de los sistemas convencionales. También se mencionan proyectos en distintas industrias, en los que se han utilizado estos programas de computadora, y que siempre han resultado muy provechosos.

³ Lenguaje de programación desarrollado como proyecto de código abierto y es administrado por la empresa Python software Foundation.

En [10], presentan una simulación de un proceso utilizando un OPC. La simulación presenta problemas de intercambio de datos y la sincronización en tiempo real; demuestran las soluciones adoptadas y dan los resultados sobre el rendimiento del OPC en este tipo de aplicaciones. Resaltan que entre las ventajas que tiene el OPC sobre otros sistemas convencionales, está la independencia del desarrollo de la simulación de los mecanismos de comunicación; otra ventaja, es el acceso a las simulaciones por una amplia gama de aplicaciones, debido a la norma prevista por el OPC; el empleo de bajo costo de equipos convencionales y el apoyo para las simulaciones a gran escala.

En [11] dice que mediante la combinación de objetos “OLE” y “OPC” para el control de procesos en tiempo real, se pueden obtener datos bastante fiables y de alta calidad. Y que es posible crear software con control como el HMI y el SCADA.

2.3 MODELOS PARA EL MONITOREO DE COSTOS EN TIEMPO REAL

Al revisar una cierta cantidad de artículos, no se encontró ninguno que hablara de algún estudio en el que se haya monitoreado el costo en tiempo real. Sin embargo, Méndez [12] desarrolló un modelo de estimación de costos al ensamblar productos electrónicos. La intención de este modelo fue poder estimar el costo de productos electrónicos nuevos en las etapas iniciales de desarrollo y diseño. Sin embargo, dicho modelo podría ser utilizado como base para los cálculos necesarios en un sistema de monitoreo de costos en tiempo real.

La empresa siempre hallará recursos que pueden ser compartidos por diferentes artículos. Los costos de estos recursos son parte del gasto indirecto que el sistema de contabilidad de costos de la compañía debe llevar. El sistema de costos basado en actividades (ABC) incrementa la exactitud del costo porque se centra en las relaciones de causa-efecto entre el trabajo ejecutado (actividades) y el consumo de recursos (costos) [13].

El objetivo del costeo basado en actividades es vigilar los costos a los productos o servicios en lugar de asignarlos de manera arbitraria. Los materiales directos y la mano de obra directa en general, se asignan a los productos, debido a que tienen una medida física de su consumo para un producto particular [14]. En el pasado, casi todas las industrias utilizaban horas de mano de obra directa como

el único factor de costos para aplicar los costos a los productos. Pero las horas de mano de obra directa no representan una medida muy apropiada de la causa de los costos en las empresas modernas, automatizadas. Los costos relacionados con la mano de obra en un sistema automatizado pueden representar tan sólo del 5% al 10% de los costos totales de producción y en muchas ocasiones no están relacionados a las causas de la mayoría de costos de gastos indirectos de producción. En consecuencia, muchas empresas están empezando a emplear las horas-máquina como su base para la asignación de costos. Otras están implementando lo contabilidad a base de actividades para desarrollar medidas especiales que reflejen de mejor manera las causas de los costos en su entorno.

En [15] comparan el método tradicional de asignar los costos, con el método ABC. Dicen que una empresa que está funcionando bien con el sistema tradicional, no es recomendable que implemente el método ABC. Benítez [16] dice que el propósito de un sistema de costos, es llegar a estimar el costo unitario de los productos. Por costo unitario entendemos la suma de todos los costos directos más una adecuada proporción de costos indirectos. El problema radica precisamente en lo que cabe entender por una proporción adecuada de los costos indirectos, actualmente se calculan los costos según el volumen de producción. El costo total así calculado es una aproximación actual del costo a largo plazo. El sistema de costos ABC se centra específicamente en la gestión de actividades como la forma que tienen las empresas para ganar competitividad.

Una de las ventajas del ABC es que se trata de una metodología bastante simple y clara. ABC apunta a hacer un mapeo preciso de los costos desde su origen hasta su destino, haciendo uso del concepto de actividad. Las actividades son las grandes tareas que se realizan dentro de una organización. Están estrechamente relacionadas con otro término muy difundido que es el de proceso. Un proceso puede ser definido como un flujo de actividades que es ejecutado con un objetivo determinado. Por ejemplo, en forma simplificada el proceso de venta puede descomponerse en las siguientes grandes actividades: Investigar el mercado, tomar pedidos, procesar pedidos, producir, entregar, facturar y cobrar.

En esta sección, se han citado algunos trabajos previos, los modelos matemáticos y la tecnología para desarrollar e implantar un sistema, a partir de esta información hace falta diseñar un sistema

real. Se encontraron varios artículos en los que se han monitoreado distintos valores en manufactura en tiempo real, pero no se encontró ninguno en el que se monitorearan los costos. En esta sección, también se definen los conceptos necesarios que se mencionan más adelante en este artículo.

3. LA FÁBRICA MODELO COMO PLATAFORMA PARA LA CREACIÓN DE UN PROTOTIPO

Para la implantación de este proyecto se eligió la Fabrica Modelo del Departamento de Ingeniería Industrial por ser un lugar accesible, el cual tenía los elementos básicos de una línea de producción moderna. En esta sección, se describen cada una de las estaciones de la línea y lo que origina costos en cada una de ellas.

El Departamento de Ingeniería Industrial en la Universidad de Puerto Rico (UPR) en Mayagüez ha establecido en uno de sus laboratorios un negocio de alta tecnología "con fines de lucro" [17]. La iniciativa ha recibido el nombre de Fábrica Modelo del RUM, e incluye una línea de montaje de componentes electrónicos a tarjetas de circuitos impresos. La Fábrica Modelo aparenta ser el lugar ideal para implantar un sistema de monitoreo de costos en tiempo real utilizando como base para los cálculos el modelo de Méndez.

La actividad, es operada y realizada por estudiantes de ingeniería supervisados por el Departamento de Ingeniería Industrial. Los estudiantes participantes deben asistir a un curso interdisciplinario en el que aprenden la tecnología de montaje superficial (SMT), que es la utilizada para ensamblar los productos en esta línea [18]. La Fábrica Modelo representa una empresa pequeña de manufactura en un ambiente académico. Como la inmensa mayoría de los negocios pequeños, tiene recursos limitados, tanto en términos de personal como económicos. Está equipada con una línea automatizada para ensamblar componentes electrónicos a tarjetas de circuitos impresos (PCB). Actualmente, en la fábrica se ensamblan tres modelos de circuitos electrónicos que, a su vez, son sub-ensambles de un producto médico. En la figura 1 está el diagrama del proceso de ensamblaje, y en la figura 2 está el despliegue de las facilidades de la Fábrica Modelo.

En [19], dicen que el Departamento de Ingeniería Industrial fomenta el espíritu empresarial entre estudiantes de distintas disciplinas. La iniciativa llamada Fábrica Modelo, ha sido beneficiosa tanto para el mundo académico, como para la industria local. Dado que este proyecto tiene una finalidad práctica donde se estudie la implantación de un sistema de monitoreo de costos en tiempo real, resulta conveniente usar la Fábrica Modelo para implantar un sistema de este tipo.

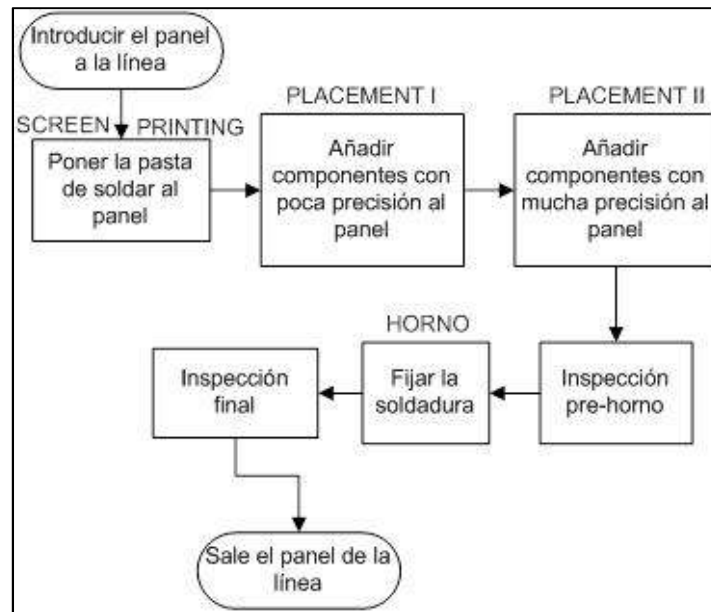


Figura 1 Diagrama del Proceso de Ensamblaje

Serigrafía.- En la estación de la figura 3 se pone la pasta de plomo en los lugares donde se van a soldar los componentes al panel (*screen printing*) se introducen los paneles uno a uno. Para el cálculo de los costos que se suman en esta estación, están el costo del plomo, el de los paneles, el de mantenimiento y el del alquiler de la máquina. Además están los costos indirectos, que se distribuyen entre todas las estaciones, entre estos están el costo de electricidad por el A/C, por la iluminación, por la electricidad que consumen otros equipos eléctricos de la fábrica, el costo del acarreo de materiales, del alquiler del edificio que consume esta máquina y la tercera parte del consumo de electricidad del compresor.

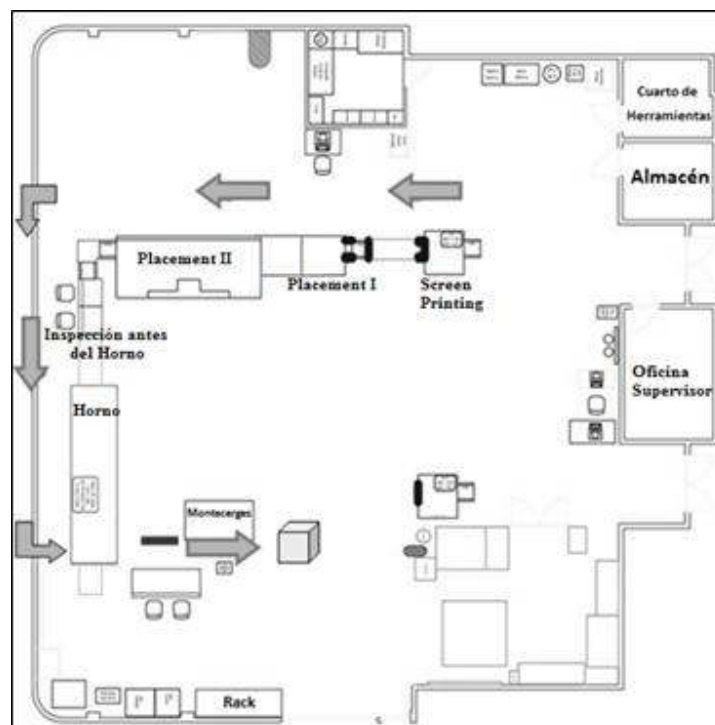


Figura 2 Despliegue de Facilidades de la Fábrica Modelo



Figura 3 Estación 1: Serigrafía

MÁQUINA DE ENSAMBLAJE I.- El panel pasa por medio de una correa transportadora a la segunda estación y entra a esta máquina (figura 4). En esta estación se colocan gran cantidad de componentes que no requieren mucha precisión (figura 5). El trabajo en esta estación es totalmente automatizado, el operario, sólo interviene para el alistamiento, para sustituir los rollos de componentes cuando se terminan y para corregir y re-arrancar la máquina cuando hay algún error en el ensamblaje de los componentes. Para el cálculo de los costos, se suman los costos de cada uno de los componentes que se ensamblan, el de mantenimiento, el de alquiler y el de consumo de la electricidad de esta máquina. Además, se suman los costos de electricidad por el A/C, por la iluminación, por el consumo de electricidad de otros aparatos, el de alquiler de la fábrica y la tercera parte del consumo de energía del compresor.



Figura 4 Estación 2: Máquina de Ensamblaje I

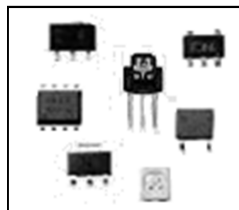


Figura 5 Componentes que ensambla la máquina de Ensamblaje I

MÁQUINA DE ENSAMBLAJE II.- El panel entra a esta máquina (figura 7). por medio de una correa transportadora. En esta estación se colocan los componentes que requieren mucha precisión (figura 6). El trabajo en esta estación es totalmente automatizado, el operario, sólo interviene para el alistamiento, para sustituir los rollos de componentes cuando se terminan y para corregir y re-arrancar la máquina cuando hay algún error en el ensamblaje de los componentes. Para el cálculo de los costos, se suman los costos de cada uno de los componentes que se ensamblan, el de mantenimiento, el de alquiler y el de consumo de la electricidad de esta máquina. Además, se le suman los costos de electricidad por el A/C, por la iluminación, por el consumo de electricidad de otros aparatos, el de alquiler de la fábrica y la tercera parte del consumo de energía del compresor.

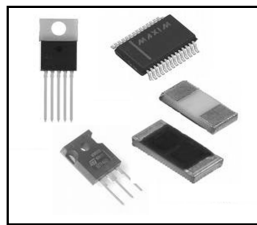


Figura 6 Componentes que ensambla la máquina de Ensamblaje II.



Figura 7 Estación 3: Máquina de Ensamblaje II

INSPECCIÓN.- En esta estación (figura 8), el operario revisa el panel y comprueba que esté bien ensamblado, el operario cuenta con un modelo del circuito que está revisando, para compararlo con el de la línea. Para el cálculo de los costos, se incluye el costo de mano de obra, pues esta estación es 100% manual. Además, se suman los costos de electricidad por el A/C, por la iluminación, por el consumo de electricidad de otros aparatos y el de alquiler de la fábrica.

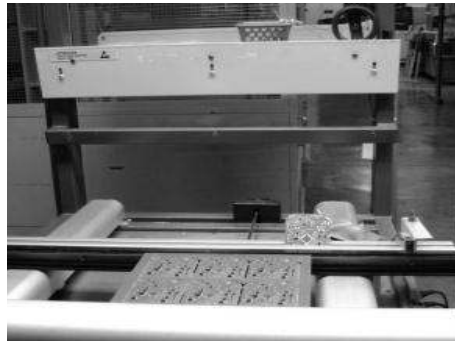


Figura 8 Estación 4: Inspección

HORNO.- Los paneles pasan a través del horno Electrovert Omniflo 7 (figura 9) donde se funde la pasta que se colocó en la primera estación, y así se sueldan los componentes al circuito impreso. Para el cálculo de los costos, se suman el del Nitrógeno que se utiliza para limpiar los paneles y que no tengan sobras de plomo, mantenimiento, el de alquiler y el de consumo de la electricidad de esta máquina. Además, se le suman los costos de electricidad por el A/C, por la iluminación, por el consumo de electricidad de otros aparatos y el de alquiler de la fábrica.



Figura 9 Estación 5: Horno

Conforme los paneles van saliendo del horno, otro operario hace una inspección final de los circuitos impresos. La inspección es muy parecida a la que se hizo antes del horno, el operario revisa que la polaridad de los componentes esté bien puesta, que estén bien alineados, que no falte ningún componente, que los componentes estén bien soldados y de ser necesario, hace algunas reparaciones menores. Para el cálculo de los costos totales de la línea en este proyecto, no se tomó en cuenta esta estación.

En esta sección, se describió la Fábrica Modelo, que fue la plataforma donde se realizó este proyecto. Se describieron cada una de las estaciones, con cuanto y de qué forma contribuye cada una con el costo final del producto. En el próximo capítulo, se va a hablar de la aplicación desarrollada de este proyecto en la Fábrica.

4. *MODELO DE COSTOS*

Las ecuaciones que se utilizaron para los cálculos de los costos reales como para los costos teóricos, están descritas en esta sección. Se utilizó el sistema ABC para calcular los costos indirectos. Para utilizar ese método, se seleccionaron las actividades que originan costos, y se individualizaron. En la tabla 1 están las actividades que suelen realizarse en una fábrica de ensamblaje de componentes en circuitos impresos.

Tabla 1 Actividades

Actividad	Factor de Costos
1. Manejo de Materiales	Número de componentes
2. Ingeniería	Hr de servicio por día
3. Preparación de Est.1	Número de prep. por lote
4. Proceso de la Est.1	Número de paneles por lote
5. Preparación de Est.2	Número de prep. por lote
6. Proceso de la Est.2	Número de paneles por lote
7. Preparación de la Est. 3	Número de prep. por lote
8. Proceso de la Est. 3	Número de comp. por imagen
9. Preparación de la Est.5	Número de comp. por imagen
10 Proceso de la Est.5	Número de paneles por lote
11 Inspección	Horas por lote

4.1 CÁLCULOS DE LOS COSTOS EN EL TIEMPO REAL

Estos cálculos, dependen de los datos recogidos de cada estación en tiempo real y de algunos costos ya suministrados. Algunos costos, se sumaron a las piezas completadas, y se dividió entre el número de unidades producidas, es decir, cada vez que salía una pieza, se iba modificando el costo unitario de todas las piezas anteriores.

Para calcular los costos, lo primero que se hizo, fue familiarizarse con el modelo Méndez, que es un modelo diseñado para este tipo de ensamblaje de circuitos impresos, este modelo es muy abarcador. Después, se seleccionaron los datos que podrían ser útiles. Seguido de eso, se analizó la producción, para seleccionar las actividades que son las que causan costos “*Cost Drivers*”. Finalmente en base a esas actividades identificadas, se fueron adjudicando los costos. En esta sección, se explican las ecuaciones que se utilizaron de acuerdo a cada actividad:

Nomenclatura:

AD	Alquiler Est.1
AE	Área Edificio (m ²)
AH	Alquiler Est.5
AIII	Alquiler Est.3
AX	Alquiler Est.2
BN	Boards por tanque de Nitrógeno
BP	Boards por pote de Plomo
CB	Cantidad de Boards
CCHIII	Costo Componentes Charger Est.3
CCHX	Costo Componentes Charger Est.2
CdeI	Costo de Ingeniería
CEACD	Costo Energía A/C Est.1
CEACI	Costo Energía A/C Est.4
CEACIII	Costo Energía A/C Est.3
CEACX	Costo Energía A/C Est.2
CEACH	Costo Energía A/C Est.5
CEBD	Costo Energía Bombillas Est.1
CEBH	Costo Energía Bombillas Est.5
CEBI	Costo Energía Bombillas Est.4
CEBIII	Costo Energía Bombillas Est.3
CEBX	Costo Energía Bombillas Est.2
CECD	Costo Energía Compresor Est.1
CECIII	Costo Energía Compresor Est.3
CECPD	Costo Energía Computadoras Est.1
CECPH	Costo Energía Computadoras Est.5
CECPI	Costo Energía Computadoras Est.4
CECPIII	Costo Energía Computadoras Est.3
CECPX	Costo Energía Computadoras Est.2
CECX	Costo Energía Compresor Est.2
CED	Costo Energía Est.1
CEH	Costo Energía Est.5
CEI	Costo Energía Est.4
CEIII	Costo Energía Est.3
CEX	Costo Energía Est.2
CMdeOI	Costo de Mano de Obra Est. 4

CMM	Costo de Manejo de Materiales
CP	Costo del Panel
CRIII	Cambio Rollo Est.3 (tiempo)
CRX	Cambio Rollo Est.2 (tiempo)
CSD	Costo de Alistamiento Est.1
CSH	Costo de Alistamiento Est.5
CSIII	Costo de Alistamiento Est.3
CSX	Costo de Alistamiento Est.2
CTD	Costo Total en la Estac. 1
CTED	Costo Total Energía Est.1
CTEH	Costo Total Energía Est.5
CTEI	Costo Total Energía Est.4
CTEIII	Costo Total Energía Est.3
CTEX	Costo Total Energía Est.2
CTH	Costo Total en la Estac. 5
CTI	Costo Total en la Estac. 4
CTIII	Costo Total en la Estac. 3
CTX	Costo Total en la Estac. 2
E	Costo del m ²
HI	Horas Ingeniero
HSD	Horas de Alistamiento Est.1
HSB	Horas de Alistamiento Est.5
HSIII	Horas de Alistamiento Est.3
HSX	Horas de Alistamiento Est.2
KWE	Kwh Energía
MD	Mantenimiento Est.1
MdeOD	Mano de Obra Directa
MDOI	Mano de Obra Ingeniero
MH	Mantenimiento Est.5
MHa	Material Handling
MIII	Mantenimiento Est.3
MOCRCHIII	\$ M. Obra Camb. de Roll. Est.3
MOCRCHX	\$ M. Obra Camb. de Roll. Est.2
MX	Mantenimiento Est.2
N	\$ Nitrógeno
NB	Número de Bombillas
NCCHIII	Num. Componentes Charger Est.3
NCCHX	Num. Componentes Charger Est.2
NE	Número de Estaciones
OD	Otros Costos Est.1
OH	Otros Costos Est.5
OIII	Otros Costos Est.3
OX	Otros Costos Est.2
PD	Paneles Diario
TD	Tiempo Est.1
TH	Tiempo Est.5
TI	Tiempo Est.4
TIII	Tiempo Est.3

TN	Tanque N (costo)
TX	Tiempo Est.2
WAC	Watts Aire Acondicionado
WB	Watts Bombilla
WC	Watts Compresor
WCP	Watts Computadoras
WD	Watts Est.1
WH	Watts Est.5
WIII	Watts Est.3
WX	Watts Est.2

1. Manejo de materiales

Se asume que el costo por el manejo de materiales es el mismo para cada componente que se ensambla. (Ec. 01).

$$CMM = MHa * CCH \quad (01)$$

2. Ingeniería

Se estimó la cantidad de horas que se utilizan al día, y se multiplicó por el costo de la hora de mano de obra del ingeniero. (Ec. 02).

$$CdeI = MDOI * HI \quad (02)$$

3. Preparación de la máquina de Serigrafía

Para distribuir el costo de alistamiento en los paneles, se sumó el costo de *alistamiento* de todas las estaciones, se sumó el costo de lo que corresponde a “Otros” de las distintas estaciones y el costo de la “Mano de obra del ingeniero”. Esos costos se pusieron una sola vez en el día, y se sumó al costo de cada una de las otras estaciones. De tal forma, que cada vez que salía un panel de la línea, se actualizaba el costo de todos los paneles anteriores. Ec. 03.

$$CSD = HSD * MdeOD \quad (03)$$

4. Proceso de la máquina de Serigrafía

El proceso de la máquina de Serigrafía, es totalmente automatizado. En esta actividad, se están considerando el costo del panel (ec. 04) y del plomo (ec. 05) que se utiliza en la estación.

$$CPP = \frac{BP}{PP} \quad (04)$$

$$CP = \frac{CB}{P} \quad (05)$$

Electricidad

Para calcular el consumo de electricidad (Ec. 11):

- a. Para la iluminación de la fábrica, se sumaron todas las bombillas del edificio, se multiplicó por la cantidad de watts que consumen, se multiplicó por el costo del Kw-hr y se multiplicó por el tiempo de proceso (Ec. 06).
- b. Un cálculo similar al realizado para obtener el costo de electricidad por panel con las bombillas, se hizo para el A/C. (Ec. 07).
- c. Se calculó el costo de electricidad utilizada por la máquina de esta estación. (Ec. 08) .
- d. El costo de electricidad del compresor, se distribuyó entre los paneles que pasan por las estaciones que utilizan ese “servicio”. Las máquinas que requieren utilizar el compresor son, la Serigrafía, las Ensamblaje I y II. (Ec. 09).
- e. La energía utilizada por las 8 computadoras que hay funcionando en la fábrica, se dividió en partes iguales a todos los paneles que se producen. (Ec. 10).

$$CEBD = \frac{(NB * WB * TD)}{1000} * KWE \quad (06)$$

$$CEACD = \frac{WAC * TD}{1000} * KWE \quad (07)$$

$$CED = \frac{WD * TD}{1000} * KWE \quad (08)$$

$$CECD = \frac{WC * TD}{1000} * KWE \quad (09)$$

$$CECPD = \frac{WCP * TD}{1000} * KWE \quad (10)$$

$$CTED = CEBD + CEACD + CED + CECD + CECPD \quad (11)$$

Otros de la máquina de Serigrafía

Otros costos, fueron el de alquiler del metro cuadrado del edificio de la fábrica, el mantenimiento y el alquiler de las máquinas. (Ec. 12)

$$OD = (AE * E) + MD + AD \quad (12)$$

5. Preparación de la máquina de Ensamblaje I

El costo de esta actividad: de alistamiento, se distribuyó igual que en la actividad 3. (Ec. 13)

$$CSX = HSX * MdeOD \quad (13)$$

6. Proceso de la máquina de Ensamblaje I

El proceso de esta máquina, es totalmente automatizado. Uno de los costos que se suman en esta actividad, es el de la mano de obra por el cambio de rollos. Se está asumiendo que cada rollo se cambia una vez al día. Conociendo cuántos componentes diferentes se ensamblan en esta estación, se multiplica esa cantidad por el tiempo que toma cambiar un rollo. Finalmente se multiplica por la cantidad que se paga por la mano de obra. Como costos de electricidad, en esta máquina, se suman los mismos que se sumaron en la estación anterior, a diferencia del costo específico de la electricidad que utiliza esta máquina. (Ec. 14)

$$MOCRCHX = CCHX * CRX * MdeOD \quad (14)$$

Electricidad

Para calcular el consumo de electricidad (Ec. 20):

- a. Para la iluminación de la fábrica, se sumaron todas las bombillas del edificio, se multiplicó por la cantidad de watts que consumen, se multiplicó por el costo del Kw-hr y se multiplicó por el tiempo de proceso. (Ec. 15)
- b. Un cálculo similar al realizado para obtener el costo de electricidad por panel con las bombillas, se hizo para el A/C. (Ec. 16)
- c. Se calculó el costo de electricidad utilizada por la máquina de esta estación. (Ec. 17)
- d. El costo de electricidad del compresor, se distribuyó entre los paneles que pasan por las estaciones que utilizan ese “servicio”. Las máquinas que requieren utilizar el compresor son, la Serigrafía, las máquinas de Ensamblaje I y II. (Ec. 18)

e. La energía utilizada por las 8 computadoras que hay funcionando en la fábrica, se dividió en partes iguales a todos los paneles que se producen. (Ec. 19)

$$CEBX = \frac{(NB * WB * TX)}{1000} * KWE \quad (15)$$

$$CEACX = \frac{WAC * TX}{1000} * KWE \quad (16)$$

$$CEX = \frac{WD * TX}{1000} * KWE \quad (17)$$

$$CECX = \frac{WC * TX}{1000} * KWE \quad (18)$$

$$CECPX = \frac{WCP * TX}{1000} * KWE \quad (19)$$

$$CTEX = CEBX + CEACX + CEX + CECX + CECPX \quad (20)$$

Otros de la máquina de Ensamblaje I

Otros costos fueron el mantenimiento y el alquiler de las máquinas. (Ec. 21)

$$OX = MX + AX \quad (21)$$

7. Preparación de la máquina de Ensamblaje II

El alistamiento de esta estación, se consideró igual que como se consideraron los alistamientos de las otras estaciones. (Ec. 22)

$$CSIII = HSIII * MdeOD \quad (22)$$

8. Proceso de la máquina de Ensamblaje II

Los costos que se consideraron, fueron los de mano de obra por el cambio de rollos, al igual que en la estación anterior. La electricidad considerada, fue igual que en las estaciones anteriores, pero cambiando el costo de electricidad propio de esta estación. (Ec. 23)

$$MOCRCHIII = CCHIII * CRIII * MdeOD \quad (23)$$

Electricidad

Para calcular el consumo de electricidad (Ec. 29):

- Para la iluminación de la fábrica, se sumaron todas las bombillas del edificio, se multiplicó por la cantidad de watts que consumen, se multiplicó por el costo del Kw-hr y se multiplicó por el tiempo de proceso. (Ec. 24)
- Un cálculo similar al realizado para obtener el costo de electricidad por panel con las bombillas, se hizo para el A/C. (Ec. 25)
- Se calculó el costo de electricidad utilizada por la máquina de esta estación. (Ec. 26)
- El costo de electricidad del compresor, se distribuyó entre los paneles que pasan por las estaciones que utilizan ese “servicio”. Las máquinas que requieren utilizar el compresor son, la Serigrafía, las máquina de Ensamblaje I y II. (Ec. 27)
- La energía utilizada por las 8 computadoras que hay funcionando en la fábrica, se dividió en partes iguales a todos los paneles que se producen. (Ec. 28)

$$CEBIII = \frac{(NB * WB * TIII)}{1000} * KWE \quad (24)$$

$$CEACIII = \frac{WAC * TIII}{1000} * KWE \quad (25)$$

$$CEIII = \frac{WD * TIII}{1000} * KWE \quad (26)$$

$$CECIII = \frac{WC * TIII}{1000} * KWE \quad (27)$$

$$CECPIII = \frac{WCP * TIII}{1000} * KWE \quad (28)$$

$$CTEIII = CEBIII + CEACIII + CEIII + CECIII + CECPIII \quad (29)$$

Otros de la máquina de Ensamblaje II

Otros costos, fueron el mantenimiento y el alquiler de las máquinas. (Ec. 30)

$$OIII = MIII + AIII \quad (30)$$

9. Preparación del Horno

Este tiempo de alistamiento se sumó a los otros, y se distribuyó al final del día entre las piezas que se hicieron en el periodo (Ec. 31).

$$CSH = HSH * MdeOD \quad (31)$$

10. Proceso del Horno

Se sumó el costo del tanque de Nitrógeno que se utiliza para limpiar los paneles (Ec. 32). La electricidad que se distribuyó a esta estación, además de incluir la de la iluminación, y la del A/C de la planta, se incluyó el costo de electricidad propio de esta estación y el debido a otros aparatos que consumen electricidad. Entre otros costos que se distribuyeron en esta estación, están el del alquiler del edificio, de las máquinas y el mantenimiento.

$$N = \frac{TN}{BN} \quad (32)$$

Electricidad

Para calcular el consumo de electricidad (Ec. 37):

- a. Para la iluminación de la fábrica, se sumaron todas las bombillas del edificio, se multiplicó por la cantidad de watts que consumen, se multiplicó por el costo del Kw-hr y se multiplicó por el tiempo de proceso (Ec. 33).
- b. Un cálculo similar al realizado para obtener el costo de electricidad por panel con las bombillas, se hizo para el A/C (Ec. 34).
- c. Se calculó el costo de electricidad utilizada por la máquina de esta estación (Ec. 35).
- d. La energía utilizada por las 8 computadoras que hay funcionando en la fábrica, se dividió en partes iguales a todos los paneles que se producen (Ec. 36).

$$CEBH = \frac{(NB * WB * TH)}{1000} * KWE \quad (33)$$

$$CEACH = \frac{WAC * TH}{1000} * KWE \quad (34)$$

$$CEH = \frac{WH * TH}{1000} * KWE \quad (35)$$

$$CECPH = \frac{WCP * TH}{1000} * KWE \quad (36)$$

$$CTEH = CEBH + CEACH + CEH + CECPH \quad (37)$$

Otros Horno

Otros costos, fueron el mantenimiento y el alquiler de las máquinas.

$$OH = MH + AH \quad (38)$$

11. Inspección manual

Esta estación tiene un gran costo de mano de obra porque un operario debe estar enfocado totalmente a esa estación (Ec. 39). La electricidad, es el consumo del A/C y de las computadoras.

$$CMdeOI = TI * MdeOD \quad (39)$$

Electricidad

Para calcular el consumo de electricidad (Ec. 43):

- Para la iluminación de la fábrica, se sumaron todas las bombillas del edificio, se multiplicó por la cantidad de watts que consumen, se multiplicó por el costo del Kw-hr y se multiplicó por el tiempo de proceso (Ec. 40).
- Un cálculo similar al realizado para obtener el costo de electricidad por panel con las bombillas, se hizo para el A/C (Ec. 41).
- La energía utilizada por las 8 computadoras que hay funcionando en la fábrica, se dividió en partes iguales a todos los paneles que se producen (Ec. 42).

$$CEBI = \frac{(NB * WB * TI)}{1000} * KWE \quad (40)$$

$$CEACI = \frac{WAC * TI}{1000} * KWE \quad (41)$$

$$CECPI = \frac{WCP * TI}{1000} * KWE \quad (42)$$

$$CTEI = CEBI + CEACI + CECPI \quad (43)$$

Para calcular los costos de cada estación, se suman los costos obtenidos en las actividades que apliquen. La Ec. 53 es la suma de todos los costos de cada una de las estaciones.

Suma de costos Estación 1

$$CTD = CMM + CPP + CP + CTED \quad (44)$$

Suma de costos Estación 2

$$CTX = (NCCHX * CCHX) + CTEX \quad (45)$$

Suma de costos Estación 3

$$CTIII = (NCCHIII * CCHIII) + CTEIII \quad (46)$$

Suma de costos Estación 4

$$CTI = CMdeOI + CTEI \quad (47)$$

Suma de costos Estación 5

$$CTH = N + CTEH \quad (48)$$

Suma de los costos totales de las estaciones

$$CTTE = CTD + CTX + CTIII + CTI + CTH \quad (49)$$

Suma de los costos de alistamiento de las estaciones

$$CTSE = CSD + CSX + CSIII + CSH \\ + MOCRCHIII + MOCRCHX \quad (50)$$

Suma de los costos de Otros de las estaciones

$$CTOE = OD + OX + OIII + OH \quad (51)$$

Suma de costos por panel

$$CP = \frac{CTTE + Cdel + CTSE + CTOE}{PD} \quad (52)$$

4.2 CÁLCULOS DE LOS COSTOS TEÓRICOS

En la computadora personal se calcularon los tiempos de proceso. A la vez que se calcularon los costos utilizando el tiempo real, se calcularon los tiempos de proceso y los costos teóricos utilizando ecuaciones del modelo Méndez. El modelo Méndez [12] es un modelo bien genérico y general, considera procesos que no están incluidos en la línea de producción de la Fábrica Modelo, sólo se van a utilizar las ecuaciones de los procesos que apliquen a este proyecto. En cada proceso de ensamblaje se toman en cuenta los costos de labor directa, de materiales y componentes, del equipo, del personal de soporte, de las utilerías y del espacio.

Una vez que se obtuvieron los tiempos de proceso de cada estación, con esos tiempos, se calcularon los costos reales por estación. Simultáneamente, se hicieron otros cálculos, por estación, para obtener los costos teóricos, utilizando las mismas ecuaciones que se utilizaron en el cálculo de los costos reales pero dejando los tiempos de operación de forma fija.

Al final, se presentó el costo total de la pieza terminada conforme va saliendo de la línea. El costo de alistamiento de cada una de las estaciones, el de cambio de rollos en las estaciones donde se puede acceder a su “*firmware*” y el de mano de obra del ingeniero, se suma al primer panel que sale, y conforme van saliendo nuevos paneles, el costo total de la pieza anterior, se suma y se divide entre el número de paneles terminados. Se utilizó ese método para incluir los costos fijos de alistamiento y el del sueldo del ingeniero.

5. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA PROTOTIPO

En este proyecto se creó un prototipo de un sistema que es capaz de recopilar y analizar información de costos de ensamblaje en tiempo real. Este prototipo se implantó en la Fábrica Modelo. Para su diseño se analizaron las estaciones de la línea de ensamblaje de circuitos impresos y se determinó cómo se podía obtener la información relevante de cada una de ellas para calcular los costos, así como el tiempo de proceso, la materia prima que se consume y los eventos que pueden suceder en cada estación y que a su vez tienen un impacto en el costo. En esta sección, se detalla el diseño de este prototipo el cual se dividió en cuatro partes: *Estaciones*, *Programación*, *Interfaz con el usuario* y *Archivo Stacomb.dat*.

5.1 CONEXIONES Y APARATOS FÍSICOS UTILIZADOS PARA LA CAPTURA DE DATOS

Para obtener los datos de la línea de producción, se clasificaron las estaciones de ensamblaje en dos: las estaciones en las que se podía acceder al “*firmware*”⁴ de las máquinas automáticas; y en las estaciones que no se tenía acceso al “*firmware*” de la máquina.

5.1.1 Estaciones en las que se Puede Acceder al “*Firmware*”

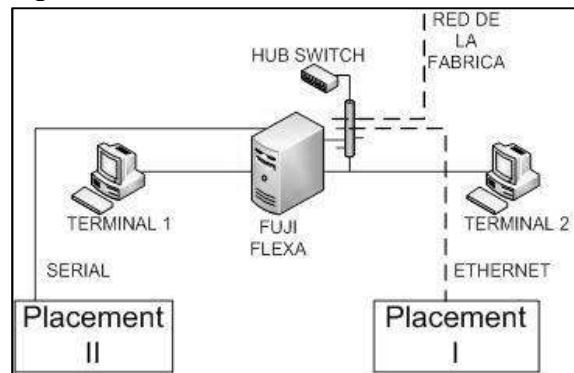


Figura 10 Esquema del cableado de las estaciones 2 y 3.

Las estaciones de ensamblaje permiten obtener cierta información a través de su “*firmware*”. Esas máquinas se pueden conectar por serial o por ethernet (figura 10).

⁴ Es una programación que está incrustada dentro de la máquina en ROM y que controla las acciones de la máquina y las comunicaciones con lo externo.

5.1.2 Estaciones en las que No se Tiene Acceso al “Firmware”

Para las estaciones donde no se podía obtener la información por medio del “firmware” de la máquina (figura 11), se determinó que harían falta dos sensores en cada una de estas estaciones para calcular el tiempo de ensamble. Se instaló un sensor a la entrada de la estación, y otro a la salida, de forma que el primero marcaba la hora de entrada del panel, y el segundo marcaba la hora de salida, y por diferencia se obtenía el tiempo de producción en esa estación. Para procesar la información que vendría de los sensores, se decidió conectarlos a un PLC modelo Direct Logic 205[®], y para programarlo, se utilizó un software llamado DirectSoft5[®], que utiliza un lenguaje de programación gráfico llamado “ladder”. El PLC, recibía las señales de todos los sensores de las distintas estaciones, y transmitía esas señales a una computadora personal.

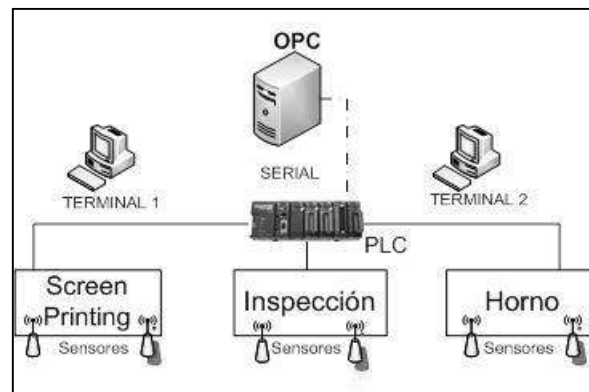


Figura 11 Esquema del cableado de las estaciones 1, 4 y 5.

El objetivo del estándar OPC es que máquinas de distintos fabricantes se puedan comunicar con facilidad con una PC, sin necesidad de crear programas especializados. Para seleccionar el OPC más adecuado para este proyecto, se estudió la página web de OPC Foundation [20] en la que explica el funcionamiento y la regulación de los OPC. Entre las distintas marcas de OPC, se identificaron los siguientes: MatrikonOPC[®], OPCKeepware[®], un OPCActiveX para C++[®], otro para VB.Net[®] y el OPCDataHub[®]. De todas esas marcas, se escogieron dos, el OPCDataHub[®] y el OPCKeepware[®], porque se vio que esos OPC eran compatibles con VBA lo cual se estaba considerando para utilizar en este proyecto. Para seleccionar entre esos dos, influyó que desde Software Toolbox OPC, la compañía que manufactura el OPCDataHub[®], dieron una licencia de 90 días de duración para su OPC con total funcionalidad y con todos los drivers que se necesitaba para las conexiones con las máquinas.

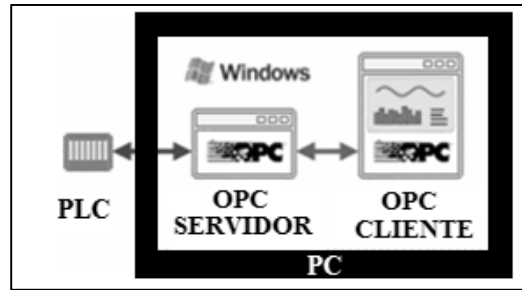


Figura 12 Conexiones del PLC a la PC por el OPC

Los sensores de la línea, fueron conectados al PLC, y éste a su vez con la computadora personal que tenía instalado el OPC. Una vez que la información estaba en el OPC, se hizo que por el protocolo *Dynamic Data Exchange* (DDE) se pasara la información a MSeExcel®. Ese protocolo (figura 12) es utilizado para la transferencia de datos entre distintas aplicaciones windows.

5.2 PROGRAMACIÓN LLEVADA A CABO PARA PROCESAR LA INFORMACIÓN DE COSTOS

El programa principal llamado “**All Stations.xlsm**” para la adquisición de datos, fue hecho en VBA®. Este programa es un ciclo que va activando y desactivando los sub-programas que recogen los datos de cada estación. Hay dos tipos de estaciones: las que se obtuvieron los datos por medio de sensores. Para esas estaciones, se hizo un módulo de programación, está explicado en la sección 5.2.1 de esta sección. El otro tipo de estaciones, son las que se podía acceder a su “*firmware*” y este módulo está explicado en la sección 5.2.2. En la figura 13 se resume toda la programación. En <https://sites.google.com/site/monitoreodecostos/apendices/Apendices.pdf?attredirects=0&d=1> están los apéndices de este documento.

5.2.1 Módulo de Adquisición de Datos en Tiempo Real de las Estaciones con PLC

Los programas correspondientes a las estaciones en las que se obtiene el tiempo de proceso por medio de sensores, fueron hechos en VBA®. El sensor localizado a la entrada de la estación marca la hora de comienzo de proceso en esa estación, y el sensor localizado a la salida de esa estación, señala que ya se terminó el proceso en esa estación. Ese tiempo de proceso, que se calcula por la diferencia entre la hora de salida menos la hora de entrada, sirve para calcular los costos reales de esa estación. Los costos teóricos se calculan con las ecuaciones, y al igual que los costos reales, se almacenan en el archivo “stacomb.dat”. Los programas correspondientes a estas estaciones, están en los apéndices respectivos.

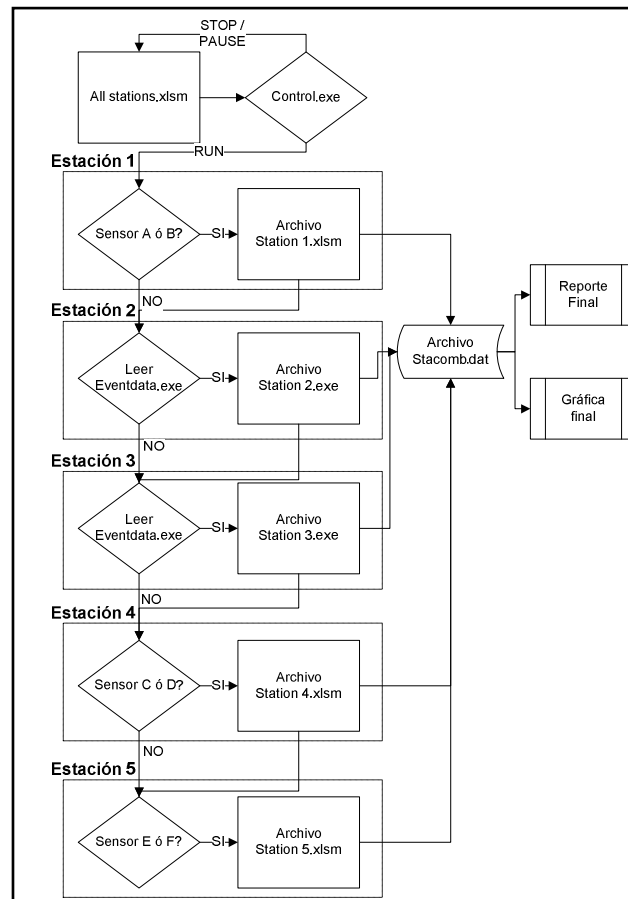


Figura 13 Diagrama del Programa

5.2.2 Módulo de Adquisición de Datos en Tiempo Real de las Estaciones a las que se Puede Accesar al “Firmware” de la Máquina

El archivo “eventdata.txt”, es un archivo que crea el programa Director de Fuji Flexa en la computadora personal. En él se registran los comandos de lo que sucede durante el ensamble de los componentes. Este archivo se va actualizando en tiempo real, posee la característica de que tiene una duración de almacenamiento de información de una hora. Cuando este archivo completa una hora de haber sido creado, los comandos anotados se reparten en distintos archivos de la computadora personal. El archivo se cierra e inmediatamente se abre otro archivo vacío con el mismo nombre y se repite lo anterior.

Los programas de las estaciones de ensamblaje, que son las máquinas a las que se puede acceder a su “firmware” fueron hechos en VB6®. Con esos programas, se leen los archivos “eventdata.txt” correspondientes a cada una de las estaciones. Estos archivos se hicieron en VB6® porque tenían

que funcionar como aplicaciones independientes de los otros programas y VB6[®] permite crear archivos ejecutables.

Los programas correspondientes a las estaciones de ensamblaje, leen el archivo “eventdata.txt” de la computadora personal que está conectada a las máquinas automáticas. Esta programación, va leyendo carácter por carácter de ese archivo, y una vez que se encuentra con un carácter que indica fin de línea, pasa a la siguiente línea.

En el programa de estas estaciones, sólo se toman en cuenta los comandos PRODSTART, ERROR y PDCOUNT. El comando PRODSTART indica cuando la máquina comienza a trabajar con un nuevo panel. El comando ERROR indica, si hay algún error en el proceso y el tipo de error. El comando PDCOUNT indica cuántos componentes se han utilizado en cada panel. Es decir, si se han utilizado la cantidad prevista, o se han utilizado más componentes de los requeridos por ese panel, si sucede eso, quiere decir que se han desperdiciado componentes. Para conocer los costos del “modelo que se ensambla”, se hicieron unos archivos *.txt. El programa de la estación correspondiente, al leer el archivo “eventdata.txt”, descubre el código del “modelo que se está ensamblando”, y el componente que se ensambla. Con esa información, busca el archivo que corresponde al “modelo que se está ensamblando”, y ahí busca el costo del componente. En la figura 14, está el diagrama de flujo del programa que obtiene la información de los archivos “eventdata.txt”. En el archivo “eventdata.txt” hay otros comandos, pero no son relevantes para este proyecto.

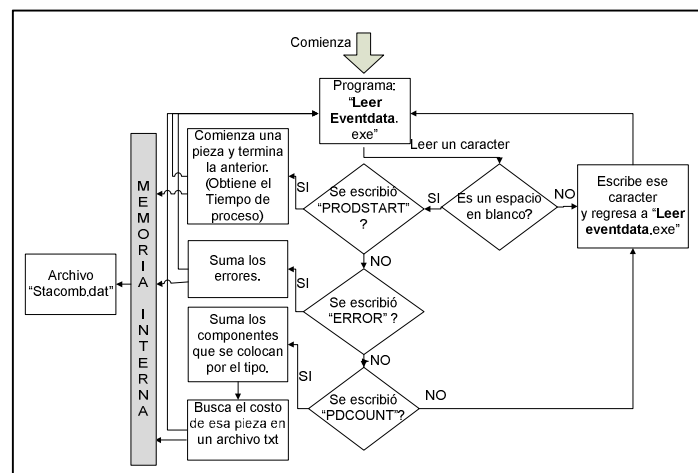


Figura 14 Diagrama de la programación para obtener datos del Archivo Eventdata.txt

5.3 INTERFAZ CON EL USUARIO

Los valores de los costos directos e indirectos que no necesariamente dependen del tiempo de proceso, son datos conocidos que se pueden suministrar al principio del proceso o dejar fijos los que ya están predeterminados. Un programa diseñado en VBA[®], pregunta si se desea cambiar alguno de los valores predeterminados de los costos. Si se desea cambiar alguno, se le dice que sí, y se siguen las instrucciones. Esta forma de introducir los datos, se hizo así, para que el usuario, no necesariamente deba tener conocimientos en programación de VBA[®].

Otro archivo, que es de interface gráfica, es el archivo “Control.exe”, el cual, controla el funcionamiento principal del programa **All Stations.xlsm**. Consiste en un código de VB6[®] que tiene una forma gráfica, con cuatro botones. Uno es el de **STOP**, otro es el de **PAUSE**, otro es el de **RUN** y otro es el de **EXIT**. Esos botones obligan a los programas de todas las estaciones, a hacer lo que les indique el botón seleccionado. En esa forma del control, hay dos líneas de datos, la línea superior indica los números de la situación de cada una de las cinco estaciones (0=RUN; 1=PAUSE; 2=PAUSE (máquina responde al control); 3=STOP); y la segunda línea, indica los últimos renglones en los que se ha puesto información en el archivo “stacomb.dat” correspondiente a cada estación.

5.4 ARCHIVO “STACOMB.DAT”

El archivo “stacomb.dat”, es un archivo de tipo aleatorio o directo, este tipo de archivos, almacena los datos en forma de registros. Los programas de cada estación suministran datos a este archivo, y es aquí donde se almacena la información que más tarde se utilizará para hacer la gráfica y el reporte final. Cada record de este archivo, tiene un grupo de 6 columnas (grupo A) y cada una de esas columnas, está dividido en 5 sub-columnas (grupo B). Las columnas 2 a la 6 del grupo A, corresponden a las 5 estaciones. Las columnas 1 a la 5 del grupo B, son para el Record Locking, Costo Teórico, Costo Real, Errores y Componentes Desperdiciados respectivamente.

En el caso de las estaciones que requieren sensores sólo utilizan las primera tres columnas del grupo B. En la cuarta sub-columna de la última estación se anota la suma de los costos reales de todas las

estaciones. En la quinta sub-columna de la última estación del último panel terminado, se anota el promedio del panel que acaba de terminar. Las primeras cinco celdas del grupo A en el record 1, tienen un uso concreto, pues ahí se almacena el estado de cada una de las estaciones según el “Control”, que es la interface con el usuario, y que tiene varios botones.

Se utilizó el archivo “stacomb.dat” por varios motivos:

1. Para agrupar la información de todas las estaciones en un solo archivo.
2. Como en este proyecto no se tiene control de cada panel en específico, se usa este archivo, para llevar un control y que la información reciente que viene de una estación en específico, ocupe el primer lugar vacío que encuentre en su respectiva columna.
3. Poder manejar todas las estaciones simultáneamente desde un mismo “Control”.

6. RESULTADOS Y TRABAJOS FUTUROS

Después de que se terminó de montar este sistema, se echó a correr y se obtuvo la gráfica final y el reporte esperado. En la gráfica se compararon los costos teóricos, con los costos reales. También se obtuvo un reporte en el que dice la cantidad de paneles que se han completado, y el costo total de cada uno de esos paneles. Como ejemplo, está la gráfica de la figura 15. En esa gráfica, se pueden observar los costos teóricos y los costos reales. El resumen de esa gráfica, se puede observar en el reporte final en la figura 16. Tanto el reporte, como la gráfica, dicen la misma información pero expresada en distintas formas. Los puntos que se grafican, son los costos reales y teóricos, asumiendo que se han terminado de ensamblar “n” número de paneles. Luego se obtiene un reporte muy general, en el que aparece la cantidad de paneles completados, y el costo que se le debe asignar a cada uno de esos paneles. Este reporte, puede ayudar a la gerencia de varias formas: en el reporte, se obtienen los costos teóricos y los reales, deberían ser muy similares el uno al otro, de no serlo, los operarios o al que le corresponda, se puede dar cuenta que algo está fallando en la producción, y como esa información es en tiempo real, pueden tomar las medidas necesarias oportunamente.

Para el cálculo de costos, no se tomaron en cuenta los costos de materiales o trabajo en proceso (WIP por sus siglas en inglés) ni los de inventarios tanto de materia prima como de producto terminado, pues esos costos no eran relevantes en la fábrica modelo. Sin embargo, esos costos, pueden ser significativos en otra fábrica en la que haya grandes acumulaciones de material, y/o donde se deba desplazar grandes distancias.

Para poder aplicar este proyecto a otra fábrica de manufactura, se requiere revisar si las estaciones de trabajo son automatizadas o no. En las estaciones automatizadas, se podrían conocer los tiempos de proceso por medio de su *firmware*, y en las estaciones a las que no se pueda acceder a su *firmware*, se requiere instalar sensores al principio y al final de la estación, luego el PLC e instalar el software necesario en la computadora que se utilizará como servidor.

Tal y como está diseñado este sistema, no se facilita hacer “*part tracking*”, para poder hacerlo, se podrían utilizar algunas herramientas tal y como poner un código de barras a cada panel; o utilizar equipos de Radio Frecuencia. Se puede conseguir que con el sistema actual se haga un “*part*

tracking”, pero es necesario contar con que cada vez que se saque una pieza de la línea, un operario le indique de alguna forma al programa de computadora.

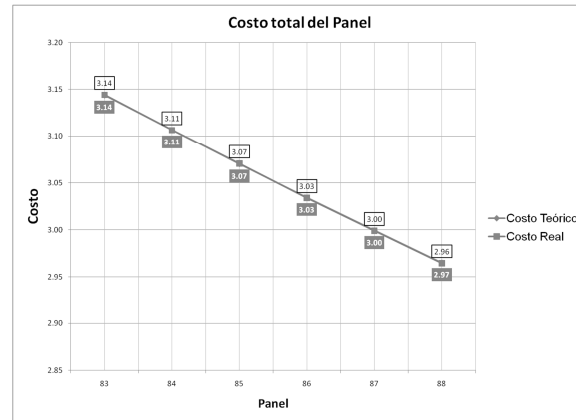


Figura 15 Gráfica final

REPORTE	
COSTO TEÓRICO DE LA PIEZA	3.07
COSTO REAL DE LA PIEZA	3.07
CANTIDAD DE PIEZAS	85

Figura 16 Reporte final de Costos

7. REFERENCIAS

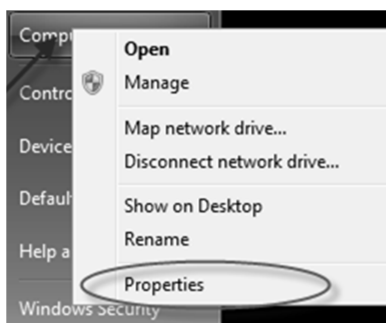
- [1] G. Frontini and S. Kennedy, "Manufacturing in Real Time". Ed. Butterworth Heinemann. 2^{da} ed. Canadá, 44 - 68. 2003.
- [2] Leadtec™. [Sitio Web]. Accesado en: 16 Apr. 2011. Disponible: <http://www.leadtec.com>.
- [3] A. Kyaw Soe, A. Naing Myint, and M. Maung Latt. "Real-Time Monitoring of Scada based Control System for Filling Process". AIP Conference Proceedings, Volume 1052, pp. 209-215. 7 Oct. 2008.
- [4] M. Zhenjun, W. Shengwei, y X. Fu. "A Supervisory Control Strategy for Building Cooling Water Systems for Practical and Real Time Applications". The Hong Kong Polytechnic University, p. 2324-2336, 2008.
- [5] AzeoTech's DAQFactory. [Sitio Web]. Accesado en: Apr. 2011. Disponible: <http://www.azeotech.com>.
- [6] A. Ramirez, S. Zhu, S.K.H. Chan, M. Ficocelli y B. Benhabib. "A Hybrid PC/PLC Architecture for Manufacturing System Control Theory and Implementation". Journal of Intelligent Manufacturing, v.13, p.261-281 2004.
- [7] K. A. Jamsa. "Welcome To Personal Computers: From Mystery to Mastery". Wonderbook_usa May 1995.
- [8] X. Hong and W. Jianhua, "Using Standard Components in Automation Indutry: A study on OPC Specification". Computer Standards & Interfaces 2005.
- [9] E. Murphy, "OPC enables interoperability". Instruments and Networks, p. 51-52. 2007.
- [10] S. R. Alves, Julio E. Normey-Rico, G. A. Merino, A. L.F. Acebes, and M. C. Prada. "OPC Based Distributed Real Time Simulation of Complex Continous Processes". Simulation Modeling Practice and Theory. p. 13. 2005.
- [11] P. Fisher, "OPC and Real-Time". Technology Update Conference, v. 2, n. 1, p. 177-186. 1998.
- [12] M. Mendez, "Development of Cost Models for Electronic Assemblies," University of Puerto Rico, Mayaguez, Tesis. 1998.
- [13] Ch. T. Horngren, G.L. Sundem, and W.O. Stratton, Pearson Education, Mexico. 2006.
- [14] L. Flores. "Costeo Abc", Facultad de Economía Universidad de Buenos Aires. Buenos Aires, 2005, pp. 2-6.
- [15] B. M. Chaffman and John Talbott, "Activity-based Costing in a Service Organization." CMA Magazine. v. 64. n. 10. p. 15. 1991.
- [16] S. Benitez. "ABC : Activity Based Costing. El Sistema de Costos ABC". Actuarios. pp 51-55. May-abr 1998.

- [17] P. Resto, D. González, A. Rullán y M. Padrón. "Establishing a Manufacturing Business in Academia" 2005.
- [18] R. Prasad, "Principles and Practice". Ray Prasad Consultancy Group, 2A ed Beaberton. ON , v. 1, 182 p. 2005.
- [19] P. Resto, D. González, A. Rullán y M. Padrón, "Entrepreneurship-Driven IE as Response to the Flat World" 2007.
- [20] Opc Foundation, [Sitio Web]. Accesado en: Jan 2010. Disponible: [http//web:opcfoundation.org](http://web:opcfoundation.org)

8. APÉNDICES

Apéndice 1 Activar/Desactivar el "Remote Desktop" en Windows 7

1. Apretar el botón de la derecha del mouse en **Computer > Properties**.

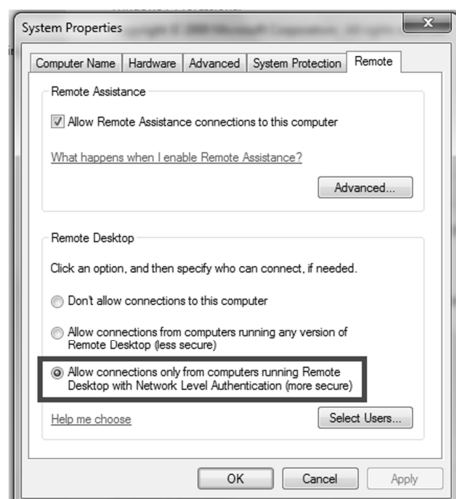


2. Del menú que aparece a la izquierda, seleccionar **"Remote settings"**.



3. En **"Remote Desktop"**, seleccione la opción que dice:

Allow connections only from computers running Remote Desktop with Network Level Authentication (more secure).

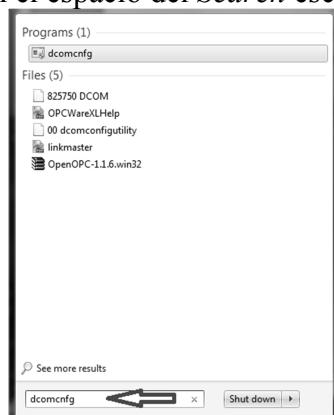


4. Si la computadora tiene “*firewall*” de Windows, el “*firewall*” se configura automáticamente para permitir el “*Remote Desktop*”; pero si la computadora tiene otro “*firewall*” es necesario configurarlo manualmente.

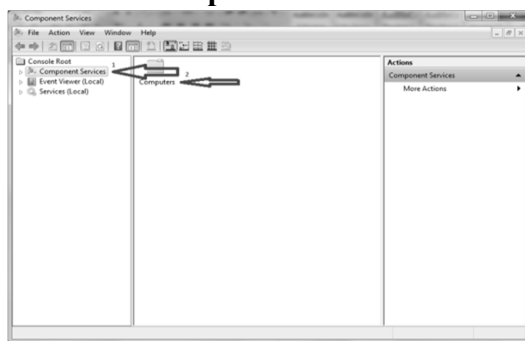
Apéndice 2 Configurar la computadora con Windows 7 para utilizar el OPC DataHub

Para utilizar el OPC en una computadora con Windows 7, se deben hacer varias configuraciones:

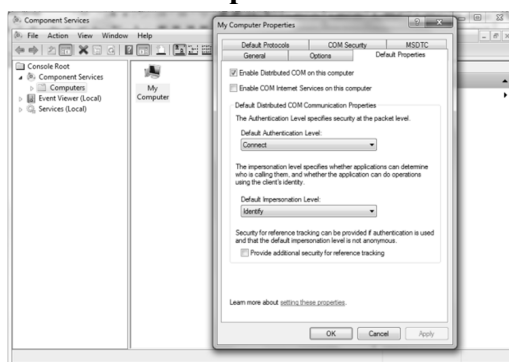
Seleccionar *START*, y en el espacio del *Search* escribir **dcomcnfg**. Seguido de “enter”.



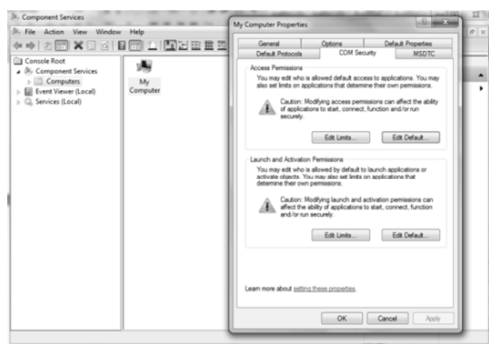
1. Se abre una ventana como la de la siguiente figura, y dar doble click en **Component Services**, doble click en **Computers**:



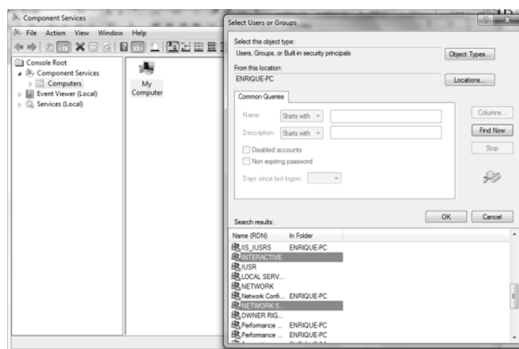
2. Oprimir el botón de la derecha del *mouse* en **My Computer**, seleccionar **Properties**, seleccionar el tab de **Default Properties** :



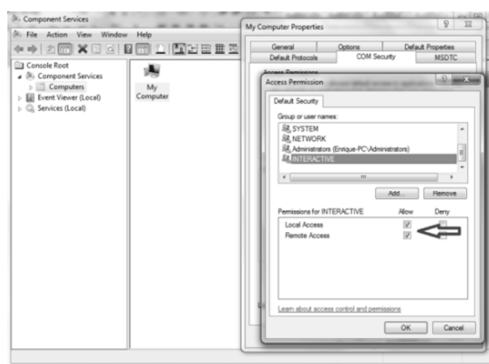
3. En Properties seleccionar el tab de COM Security, seleccionar Edit Default, Add, Advanced, Find now.



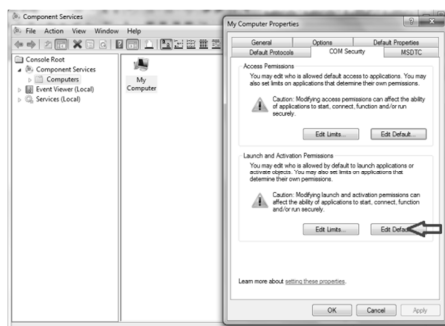
4. En Select Users groups, seleccionar: "Everyone", "Interactive", "Network" y "System" luego oprimir OK.



5. En **Access permission**, revisar que los cuatro grupos de usuarios seleccionados, tengan "allow" en las conexiones locales y remotas.



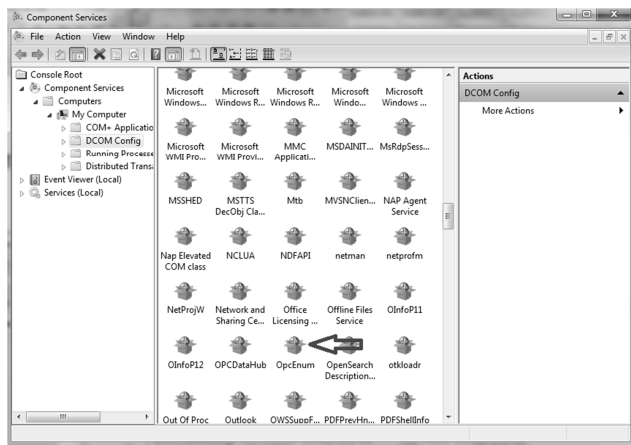
6. Repetir pasos 5 y 6 seleccionando **Launch and activation permission:**



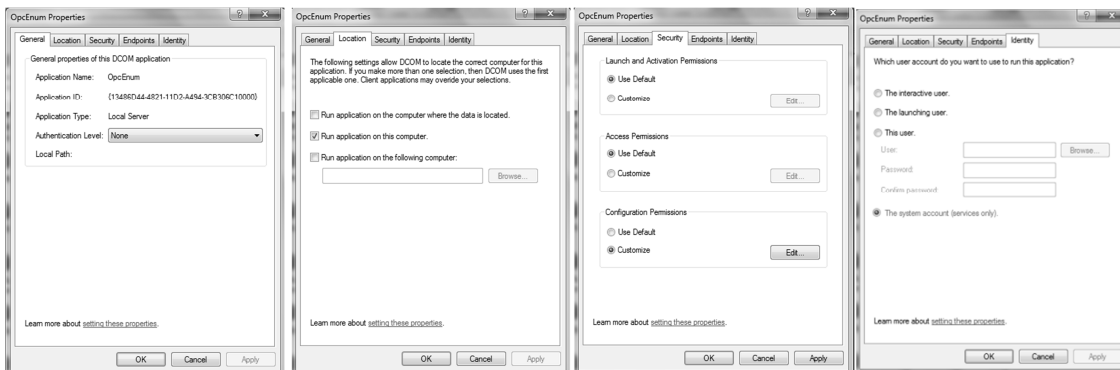
7. Repetir los pasos 4 al 7 con **Edit limits**.

Configuración del OPCENUM

1. En dcomcnfg, seleccionar Components Services > Computers > My Computer > DCOM Config > OpcEnum



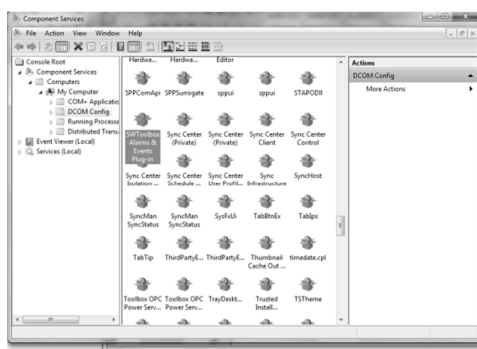
2. En **OpcEnum** oprimir right click, **Properties** y seleccionar lo que dicen las figuras.



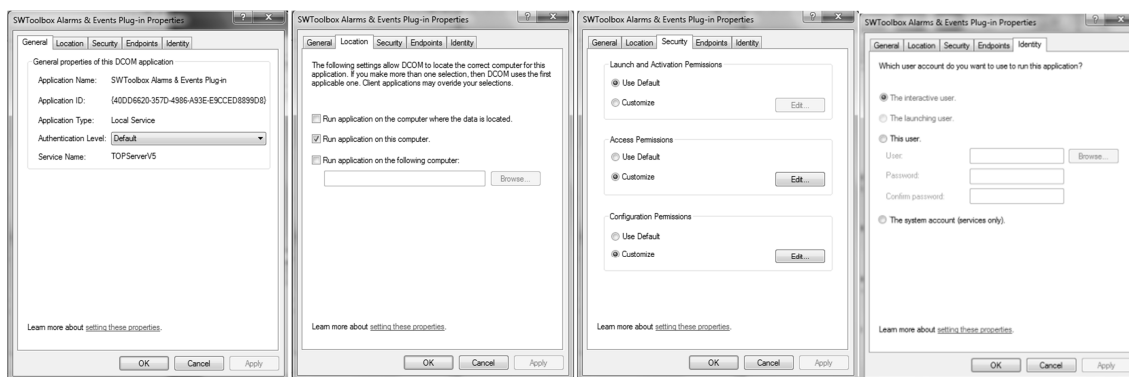
3. En Security > Configuration Permissions > Edit > Edit Default, Add, Advanced, Find now. En Select Users groups, seleccionar: “Everyone”, “Interactive”, “Network” y “System” luego oprimir OK.
4. En Change configuration permission revisar que los cuatro grupos de usuarios seleccionados, tengan “allow”. OK

Configuración del OPC Server

1. En el archivo dcomcnfg, seleccionar Components Services > Computers > My Computer > DCOM Config > SWToolbox Alarms & Events Plug-in



2. En SWToolbox Alarms & Events Plug-in oprimir right click, **Properties** y seleccionar lo que dicen las figuras.



3. En Security > Configuration Permissions > Edit > Edit Default, Add, Advanced, Find now. En Select Users groups, seleccionar: “Everyone”, “Interactive”, “Network” y “System” luego oprimir OK.
4. En Change configuration permission revisar que los cuatro grupos de usuarios seleccionados, tengan “allow”. OK

Apéndice 3 Módulo "Datoscostos.vba". Interface con el usuario

DATOS PARA CALCULO DE COSTOS

Global xx

Global Msg As String

Global AreaEdificio As Double

Global BoardsN As Double

Global BoardsPlomo As Double

Global BotePlomo As Double

Global CambioRolloIPIII As Double

Global CambioRolloXP143 As Double

Global CantidadBoards As Double

Global ComponentesCharger As Double

Global ComponentesChargerIPIII As Double

Global ComponentesChargerXP143 As Double

Global CostoEnergiaDEK As Double

Global CostoEnergiaHORNO As Double

Global CostoEnergiaINSPECCION As Double

Global CostoEnergiaIPIII As Double

Global CostoEnergiaXP143 As Double

Global CostoHORNO As Double

Global CostoRawMDEK As Double

Global CostosComponentes As Double

Global CostosINSPECCION As Double

Global CostosIPIII As Double

Global CostosXP143 As Double

Global CostoTotalDEK As Double

Global CostoTotalHORNO As Double

Global CostoTotalINSPECCION As Double

Global CostoTotalIPIII As Double

Global CostoTotalXP143 As Double

Global Edificio As Double

Global HorasIngeniero As Double

Global HrSetupDEK As Double

Global HrSetupHORNO As Double

Global HrSetupIPIII As Double

Global HrSetupXP143 As Double
 Global ImagenesBoard As Double
 Global KwEnergia As Double
 Global Lote As Double
 Global MantDEK As Double
 Global MantHORNO As Double
 Global MantIPIII As Double
 Global MantXP143 As Double
 Global MdeOD As Double
 Global MdeOIngeniero As Double
 Global MH As Double
 Global NumBombillas As Double
 Global NumComputadoras As Double
 Global NumEstaciones As Double
 Global Panel As Double
 Global PanelesDiario As Double
 Global TanqueN As Double
 Global TiempoDEK As Double
 Global TiempoHORNO As Double
 Global TiempoINSPECCION As Double
 Global TiempoIPIII As Double
 Global TiempoXP143 As Double
 Global WAC As Double
 Global WBombillas As Double
 Global WCompresor As Double
 Global WComputadoras As Double
 Global WDEK As Double
 Global WHORNO As Double
 Global WIPIII As Double
 Global WXP143 As Double

Sub Datos_costos()

xx = 1

Dim Msg As String

'Estos valores solo sirven para la simulacion

TiempoDEK = 0.08

TiempoHORNO = 0.03

TiempoINSPECCION = 0.12
TiempoIPIII = 0.08
TiempoXP143 = 0.1
AreaEdificio = 3000
BoardsN = 5000
BoardsPlomo = 1000
BotePlomo = 450
CambioRolloIPIII = 0.083
CambioRolloXP143 = 0.033
CantidadBoards = 1000
ComponentesCharger = 61
ComponentesChargerIPIII = 44
ComponentesChargerXP143 = 17
Edificio = 100
HorasIngeniero = 2
HrSetupDEK = 0.1
HrSetupHORNO = 0.05
HrSetupIPIII = 0.5
HrSetupXP143 = 0.5
ImágenesBoard = 6
KwEnergia = 0.35
Lote = 48
MantDEK = 100
MantHORNO = 100
MantIPIII = 100
MantXP143 = 100
MdeOD = 7.25
MdeOIngeniero = 60
MH = 0.1
NumBombillas = 112
NumComputadoras = 8
NumEstaciones = 5
Panel = 520
PanelesDiario = 240
TanqueN = 200
WAC = 12000
WBombillas = 40

WCompresor = 750

WComputadoras = 250

WDEK = 3000

WHORNO = 50000

WIPIII = 4500

WXP143 = 4000

Msg = "Quiere introducir nuevos valores de costos y de tiempos? Oprima Yes. Si no quiere cambiar los valores que ya tiene, oprima No."

'Display MessageBox

Answer = MsgBox(Msg, vbQuestion + vbYesNo, "Datos para calcular Costos")

If Answer = vbNo Then Module6.Estaciones

If Answer = vbYes Then

'Costo del Bote de plomo \$450

BotePlomo = InputBox("Cuál es el costo del bote de plomo? (en U\$)", , 450)

'Costo de 1000 Paneles \$520.00

Panel = InputBox("Cuál es el costo del panel? (en U\$)", , 520)

'Rendimiento 1,000 boards

BoardsPlomo = InputBox("Cuantos boards alcanza un bote de plomo?", , 1000)

'Cantidad 1,000 boards

CantidadBoards = InputBox("Cuantos boards tiene un lote de boards?", , 1000)

'Costo del KW hora \$0.35 \$/hr

KwEnergia = InputBox("Cuál es el costo del KwV por hora? (en U\$)", , 0.35)

'\$100 /pie cuadrado

Edificio = InputBox("Cuál es el costo del pie cuadrado de edificio? (en U\$/mes)", , 100)

'3,000 pies cuadrados aprox

AreaEdificio = InputBox("Cuál es el área del edificio? (pies cuadrados)", , 3000)

'\$200 tanque de 30 (estimado)

TanqueN = InputBox("Cuál es el costo del tanque de Nitrógeno? (en U\$)", , 200)

'5,000 boards

BoardsN = InputBox("Cuántos boards alcanza el tanque de Nitrógeno?", , 5000)

'\$7.25 /hr

MdeOD = InputBox("Cuánto cuesta la mano de obra directa? (en U\$/mes)", , 7.25)

'\$15 / hr

MdeOI = InputBox("Cuánto cuesta la mano de obra del supervisor? (en U\$/mes)", , 15)

'\$60 / hr

MdeOIngeniero = InputBox("Cuánto cuesta la mano de obra del ingeniero? (en U\$/mes)", , 60)

'2

HorasIngeniero = InputBox("Cuántas horas del ingeniero se usan al día? (en U\$/mes)", , 2)
 AlquilerDEK = InputBox("Cuál es el costo del alquiler de la máquina DEK? (en U\$/mes)", , 500)
 AlquilerXP143 = InputBox("Cuál es el costo del alquiler de la máquina XP143? (en U\$/mes)", , 500)
 AlquilerIPIII = InputBox("Cuál es el costo del alquiler de la máquina IPIII? (en U\$/mes)", , 500)
 AlquilerHORNO = InputBox("Cuál es el costo del alquiler del HORNO? (en U\$/mes)", , 500)
 '\$0.10 /componente (Estimado)
 MH = InputBox("Cuánto cuesta el manejo de materiales? (en U\$/componente)", , 0.1)
 '6 minutos
 HrSetupDEK = InputBox("Cuántos minutos toma hacer el alistamiento de la DEK? (hr/día)", , 0.1)
 '30 minutos
 HrSetupXP143 = InputBox("Cuántos minutos toma hacer el alistamiento de la XP143? (hr/día)", , 0.5)
 '30 minutos
 HrSetupIPIII = InputBox("Cuántos minutos toma hacer el alistamiento de la IPIII? (hr/día)", , 0.5)
 '3 minutos
 HrSetupHORNO = InputBox("Cuántos minutos toma hacer el alistamiento del HORNO? (hr/día)", , 0.05)
 MantDEK = InputBox("Cuánto cuesta el mantenimiento de la DEK? (en U\$/mes)", , 100)
 MantXP143 = InputBox("Cuánto cuesta el mantenimiento de la XP143? (en U\$/mes)", , 100)
 MantIPIII = InputBox("Cuánto cuesta el mantenimiento de la IPIII? (en U\$/mes)", , 100)
 MantHORNO = InputBox("Cuánto cuesta el mantenimiento del HORNO? (en U\$/mes)", , 100)
 '250W
 WComputadoras = InputBox("Cuántos watts consume una computadora? (en U\$/mes)", , 250)
 '8 computadoras
 NumComputadoras = InputBox("Cuántas computadoras hay en la fábrica?", , 8)
 '12,000 watts/h
 WAC = InputBox("Cuántos Watts consume el A/C?", , 12000)
 '3,000 watts/h
 WDEK = InputBox("Cuántos Watts consume la DEK?", , 3000)
 '4,000 watts/h
 WXP143 = InputBox("Cuántos Watts consume la XP143?", , 4000)
 '4,500 watts/h
 WIPIII = InputBox("Cuántos Watts consume la IPIII?", , 4500)
 '50,000 watts/h
 WHORNO = InputBox("Cuántos Watts consume el HORNO?", , 50000)
 '750 watts/h
 WCompresor = InputBox("Cuántos Watts consume el Compresor?", , 750)
 '40 watts/h
 WBombillas = InputBox("Cuántos Watts consume cada bombilla?", , 40)

```

'112 bombillas
NumBombillas = InputBox("Cuántas bombillas hay en la fábrica?", , 112)
'48 paneles
Lote = InputBox("Cuántos paneles forman un lote?", , 48)
'240 paneles al día
PanelesDiario = InputBox("Cuántos paneles se trabajan al día?", , 240)
'Componentes de Charger 61
ComponentesCharger = InputBox("Cuántos componentes necesita el Charger?", , 61)
'Componentes de Charger en XP143 44
ComponentesChargerXP143 = InputBox("Cuántos componentes necesita el Charger en XP143?", , 44)
'Tiempo que toma cambiar un rollo en XP143 0.033
CambioRolloXP143 = InputBox("Tiempo que toma cambiar un rollo en XP143?", , 0.033)
'Componentes de Charger en IPIII 17
ComponentesChargerIPIII = InputBox("Cuántos componentes necesita el Charger en IPIII?", , 17)
'Tiempo que toma cambiar un rollo en IPIII 0.083
CambioRolloIPIII = InputBox("Tiempo que toma cambiar un rollo en IPIII?", , 0.083)
'Imagenes por board? 6
ImagenesBoard = InputBox("Cuántas imágenes hay por board?", , 6)
'Numero de estaciones? 5
NumEstaciones = InputBox("Cuántas estaciones hay en la línea?", , 5)
End If
If xx <> "" Then Module6.Estaciones
End Sub

```

Apéndice 4 Módulo “Todas las estaciones”

Este archivo hace funcionar cíclicamente los módulos de todas las estaciones, de forma que nunca están funcionando simultáneamente dos estaciones.

```
'=====
'
'                                TODAS LAS ESTACIONES
'
'=====

Sub Estaciones()

    For yy = 1 To 500
        If xx = 1 Then Module1.Form_Load
        If xx = 2 Then GoTo Line2
        If xx = 3 Then GoTo Line3
        If xx = 4 Then GoTo Line4
        If xx = 5 Then GoTo Line5
        Call Module2.Datos_costos
        Call Module1.Form_Load 'Estacion 1
Line2:
        Call Shell("C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\febrero 2011\Final\Project 2-40.exe", 1) 'Estacion 2
Line3:
        Call Shell("C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\febrero 2011\Final\station 3-v60.exe", 1) 'Estacion 3
Line4:
        Call Module4.Form_Load 'Estacion 4
Line5:
        ' Call Module5.Form_Load 'Estacion 5
        'Call Module9.GenerarDataChart3 'Graficar estacion 3
        Next yy
    End Sub
```

Apéndice 5 Módulos “Station 2.exe y Station 3.exe”

Los módulos “station 2.exe” y “station 3.exe”, fueron creados en VB6®. Esto se hizo así, porque el VBA® no puede hacer funcionar 2 módulos o más a la vez, y aquí se requería que los módulos correspondientes a todas las estaciones, estén activos simultáneamente. Después de programarlos en VB6®, se convirtieron a aplicaciones Windows que pueden funcionar independientemente uno del otro.

El archivo “station 2.exe”, va leyendo los datos del archivo eventdata.txt de la computadora, que pertenece a la estación de la máquina XP143 y el archivo “station 3.exe”, va leyendo los datos de otro archivo también llamado eventdata.txt de la computadora, que pertenece a la estación de la máquina IP III.

La información que interese almacenar del archivo “eventdata.txt” se utiliza para calcular los costos. Según el número de pieza, se busca en una lista el costo total sumando los costos de los otros conceptos. Esa información se almacena en el archivo “stacomb.dat”.

Los siguientes son los códigos correspondientes a “station 2.exe” y a “station 3.exe”:

Programa “station 2.exe”

```
Private Type t_DataRec
sta(5, 4) As Single
End Type
Sub Form_Load()
Dim DataRec As t_DataRec
Dim nada As String
Dim x As String * 1 'se declara de 1 sólo caracter, para poder leer el
                    'archivo caracter a caracter
Dim LineaDeCaracteres As String 'variable donde se irá construyendo la línea
                    'de caracteres que se leera uno a uno del
                    'archivo.
Dim filename As String    'nombre del archivo a procesar (eventdata)
Dim filename1 As String   'nombre del archivo a procesar (stacomb)
Dim fsize As Long         'tamaño de ese archivo
Dim fpos As Long          'posición del ultimo caracter leído
Dim DateTimeStamp As String
```

```

Dim DateStamp As String
Dim DateStampSav As String
Dim TimeStamp As String
Dim TimeStampSav As String
Dim ProductID As String
Dim ProductIDSav As String
Dim TimeForProdMin As Single
Dim ProductSlot As Integer
Dim ProdSlotTable(10, 60, 2) As Single
Dim ProductSlot2 As Integer
Dim ProdSlotTable2(10, 60, 2) As Single
Dim ProdSlotID(10) As Long
Dim ProdSlotID2(10) As Long
Dim ErrorCount As Integer
Dim CompNum As Integer
Dim ExtraComp As Integer
Dim SlotNum As Integer
Dim PrimeraLinea As String
Dim Linea As String
Dim recptr As Integer

```

```

'=====
'
'                                STATION 2
'=====

DoEvents
nada = ""
recptr = 1
swPrimeraLinea = 0
PrimeraLinea = ""
fsize = 0
swProdStart = 0
On Error Resume Next
'esta sección se encarga de cotejar si el archivo a leer tiene datos
'en él para procesar, usando como criterio su tamaño. Se crea un loop que
'se repetira, gastando un segundo en cada vuelta
'hasta que aparezca que hay datos en el mismo.
'cp ESS

```

```

filename = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\eventdata2.txt"
'Fuji Flexa cp
'filename = "C:\FujiFlexa\Server\Data\Prod\Xp143\eventdata.txt"
getsize:
fsize = FileLen(filename)
'si el if NO se cumple (hay datos), se sigue abajo y pasamos a abrir
'el archivo
If fsize < 1 Then
For er = 1 To 100
GoSub Waste1      'ir a perder un segundo, para dar tiempo a que
                    'el archivo aparezca con datos
Next er
End If
'GoTo getsize
'se abre el archivo de texto como si fuera un random file, del cual
'se van a leer records de UN caracter de largo
Open filename For Random As #1 Len = 1
fpos = Seek(1) 'fpos se usa para guardar la posición del caracter
            'que se va a leer del archivo
'cp ESS
filename1 = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\stacombnew.dat"
'Fuji Flexa pc
'filename1 = "C:\FujiFlexa\Server\Data\Prod\IP3RUM\stacombnew.dat"
Open filename1 For Random As #3 Len = Len(DataRec)
'el random file que acabamos de abrir, debe haberse creado con otro programa
'que crea un archivo de 300 records. Las primeras cinco posiciones (20 bytes)
'del record 1 y del record 300 son datos de control. Después siguen 5 grupos
'de 5 posiciones cada una, un grupo para cada estación. La primera posición
'de cada grupo se utilizará para record locking, las otras cuatro para
'escribir los datos de salida: tiempo, componentes, errores y comp. extras
'Los datos de control en el record 1 son para indicar a los programas
'que procesen, pausen o dejen de ejecutar. Los datos de control del
'record 300 son para ir registrando el último record que han escrito cada
'programa de estaciones.
'-----
replloop = 1
SumCompCost = 0

```

```

sumExtraComp = 0
ErrorCount = 0
While (reploop)
'fsize = 0 'nos preparamos para el error file no found que puede darse
    'más abajo cuando estemos procesando los eof
swbad = 0 'switch que indica caracter malo leído(p.e. end of file)
LineaDeCaracteres = "" 'variable donde se construirá la línea de caracteres
    'leída. La limpiamos en cada vuelta
'aquí empieza el loop de leer los caracteres de una línea
rep = 1 'esta variable mantiene el while dando ciclos si es uno, o
    'lo acaba si es cero (true or false)
While (rep) 'loop que se repetirá hasta que hayamos leído todos los
    'caracteres que componen una línea del archivo.
    getit:
    Get #1, , x
    If Asc(x) = 10 Then GoTo getit 'si es un carácter line feed (chr(10)) lo
        ' ignoramos y leemos otro
    Debug.Print x;
    fpos = Seek(1)
    swbad = 0 'limpiamos este indicador de carácter malo
    'Mirar si podemos continuar procesado
    GoSub CheckControl
    'en cuanto el flag sea 0, hemos salido de pausa, por lo que seguiremos adelante
    '-----
    'Atrapar los caracteres que puedan indicar fin de archivo (eof).
    'Esto se puede dar en medio del archivo mientras se está escribiendo
    'o al final del archivo, cuando ya está cerrado
    If Asc(x) = 26 Or Asc(x) = 0 Then
        Seek #1, fpos - 1 'volver a posicionar el archivo en el carácter malo
        GoSub Waste1 'esperar un segundo (para dar oportunidad de que se
            'escriba un carácter bueno
        If (swPrimeraLinea) Then
            denuevo:
            Close 1
            GoSub CheckControl
            fsize = 0
            fsize = FileLen(filename)

```

```

If fsize < 1 Then GoTo denuevo 'si ha habido error file not found
    'reintentar abrirlo
Open filename For Input As #1
Line Input #1, Linea
If EOF(1) Then GoTo denuevo 'la linea no estaba completa, reintentar
Close 1
Open filename For Random As #1 Len = 1
If Linea <> PrimeraLinea Then
PrimeraLinea = Linea
fpos = 1 'si las lineas no son iguales es que es un archivo nuevo, y
    'por tanto leo desde el principio (fpos=1). Si no es archivo
    'nuevo me quedo con el fpos que traía de la lectura del archivo y
    'sigo desde ahí.
End If
Seek #1, fpos - 1 'tengo que volver a posicionar el archivo en el caracter
    'malo, porque si pasé por el line input me lo varió
End If
swbad = 1 'marcar que hemos leído un caracter malo, para no añadirlo
    'a la línea que estamos construyendo
End If
'-----
'el bloque de if que sigue se ejecutara si no es un caracter malo.
'Aquí es donde se construye la línea con los caracteres que estamos leyendo
If swbad = 0 Then
'si el caracter leído es chr(13) [RETURN, indica fin de línea], no añadirlo
'a la línea y salir para procesar la línea completa
If x = Chr(13) Then rep = 0 Else LineaDeCaracteres = LineaDeCaracteres & x
End If
Wend 'aqui acaba el ciclo de lectura caracter a caracter, para una linea
'-----
'Sección de procesar la línea leída
'cotejar si se ha guardado la primerísima linea leída
'si no, guardarla y marcar que se tiene guardada
If swPrimeraLinea = 0 Then
PrimeraLinea = LineaDeCaracteres
swPrimeraLinea = 1
End If

```

```

'procesar la línea de caracteres
If InStr(1, LineaDeCaracteres, "PRODSTART") Then GoSub ProdStart
If InStr(1, LineaDeCaracteres, "PDCOUNT") Then GoSub PdCount
If InStr(1, LineaDeCaracteres, "ERROR") Then GoSub ProError
Wend      'wend de todo el proceso, que incluye reabrir el archivo
fin:
Close
End

'subrutina que gasta (waste) un segundo
Waste1:
Debug.Print Estacion2
GoTo denuevo
End

'=====

ProdStart:
'extraer la fecha
P1 = InStr(1, LineaDeCaracteres, ":")
P2 = InStr(P1 + 1, LineaDeCaracteres, ":")
DateTimeStamp = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)
DateStamp = Left(DateTimeStamp, 8)
TimeStamp = Right(DateTimeStamp, 8)
'darle formato normal a la fecha
DateStamp = Mid(DateStamp, 5, 2) & "/" & Right(DateStamp, 2) & "/" & Left(DateStamp, 4)
'extraer el product ID completo y el número solamente (6 dígitos)
P1 = InStr(1, LineaDeCaracteres, Chr(9))
P1 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
P1 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
P2 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
ProductID = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)
ProductIDNum = Left(ProductID, 6)
'Como el tiempo total para producir un producto se toma desde el inicio de
'un producto hasta el inicio del próximo, la primera vez que se procesa
'un PROSTART no hacemos cálculo de tiempo. Guardamos los valores obtenidos
'hasta el próximo PRODSTART.
If swProdStart = 0 Then
    TimeStampSav = TimeStamp
    DateStampSav = DateStamp

```

```

End If
'Las veces subsiguientes, el encontrar un
'PRODSTART (entrar en esta subrutina) señala que debemos hacer el cálculo
'y producir una línea de resultados, que es lo que se hace en el siguiente
'bloque de IF.
If swProdStart = 1 Then
difh = Left(TimeStamp, 2) - Left(TimeStampSav, 2)
difm = Mid(TimeStamp, 3, 2) - Mid(TimeStampSav, 3, 2)
difs = Mid(TimeStamp, 5, 2) - Mid(TimeStampSav, 5, 2)
difmili = Right(TimeStamp, 2) - Right(TimeStampSav, 2)
TimeForProdMin = difh * 60 + difm + difs / 60 + difmil / 6000
GoSub DataWrite
'If nada = "" Then nada = InputBox("ENTER para seguir")
Print #4, TimeForProdMin * 2, TimeForProdMin, ProductID, SumCompCost
SumCompCost = 0
sumExtraComp = 0
ErrorCount = 0
TimeStampSav = TimeStamp
End If
swProdStart = 1
Return
'-----
ProError:
If swProdStart = 0 Then Return
ErrorCount = ErrorCount + 1
Return
'-----
PdCount:
Debug.Print "pdcount"
If swProdStart = 0 Then Return
'extraer el numero de componentes y de esos, los extras que hubo que usar
P1 = InStr(1, LineaDeCaracteres, Chr(9))
For k = 1 To 5
P1 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
Next k
P2 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
ProductSlot = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)

```

```

GoSub GetSlotCost
P1 = P2
P1 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
P2 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
CompNum = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)
P1 = P2
P2 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
ExtraComp = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)
sumExtraComp = sumExtraComp + ExtraComp
SumCompCost = SumCompCost + CompNum * SlotCost
SumCompCost2 = SumCompCost + sumExtraComp + ExtraComp * SlotCost
Debug.Print SumCompCost
Debug.Print SumCompCost2
Return
'-----
GetSlotCost:
'Buscar el product ID Num en la tabla
swfound = 0
For i = 1 To 10
If ProdSlotID(i) = ProductIDNum Then
    swfound = i
    GoTo GetCost
End If
Next i
'pc ess
slotsfilename = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\" & ProductIDNum & ".txt"
'pc fujiflexa
'slotsfilename = "C:\FujiFlexa\Server\Data\Prod\ip3rum\" & ProductIDNum & ".txt"
Open slotsfilename For Input As #2
'Buscar dimensión vacia en el array
i = 1
While (ProdSlotID(i))
i = i + 1
Wend
'Cargar los datos de la tabla en cuestion
ProdSlotID(i) = ProductIDNum
j = 0

```

```

While Not EOF(2)
    j = j + 1
    Input #2, ProdSlotTable(i, j, 1)
    Input #2, dummy
    Input #2, ProdSlotTable(i, j, 2)
Wend
SlotLines = j
GetCost:
'seleccionar el costo del slot
SlotCost = 0
j = 1
While (ProdSlotTable(i, j, 1))
If ProductSlot = ProdSlotTable(i, j, 1) Then
SlotCost = ProdSlotTable(i, j, 2)
Close 2
Return
End If
j = j + 1
Wend
Close 2
Return
'-----
DataWrite:
'Aqui se implementa record locking que dada la naturaleza del proceso
'(pasa un tiempo entre que una estación escriba y escriba la siguiente)
'me parece innecesario. Pero se deja como un ejercicio de programación
'y por si alguna vez se modifica los programas para hacer cosas con
'el archivo donde haya riesgo que un programa escriba datos a la vez que
'otro actualiza.
'primero asegurar que no hay otro programa escribiendo
'en el record, o que estamos en pausa.
Seek #3, recptr
Get #3, , DataRec
tlock = 0
For il = 1 To 5
tlock = tlock + DataRec.sta(il, 0)
Next il

```

```

'si la suma (tlock) no es cero, hay alguien usando el record
'volver a leerlo hasta que quede libre
If tlock < 0 Then GoTo DataWrite
'poner la señal de que queremos usar el record
DataRec.sta(3, 0) = 1
Seek #3, recptr
Put #3, , DataRec
Seek #3, recptr
Get #3, , DataRec
'*****3 por 1

'Columna 11 a 14 porque son las 3eras 5 columnas
'DataRec.sta(2, 1) = TimeForProdMin

DataRec.sta(2, 2) = SumCompCost + 0.0002 + 0.0017 + 0.021 + 0.171 + (MdeOD / (ComponentesChargerXP143 *
CambioRolloXP143 * PanelesDiario)) + ((AreaEdificio * Edificio) / (PanelesDiario * NumEstaciones))

costos:
'Cálculo de costos de la XP143 Teorico(estación 2)

'Alistamiento ($/panel)          'Componentes ($/panel) (el programa ya me da el total del costo
multiplicado por el num de componentes que se usan por panel 'Panel ($/panel) Área

CostosXP143 = ((HrSetupXP143 / PanelesDiario) * MdeOD) + (SumCompCost) + (MdeOD /
(ComponentesChargerXP143 * CambioRolloXP143 * PanelesDiario)) + ((AreaEdificio * Edificio) / (PanelesDiario
* NumEstaciones))

'Consumo de energía de las bombillas ($)          'Consumo de energía del A/C ($)
'Consumo de energía de la XP143 ($)          'Consumo de energía del compresor ($)          "Consumo de energía
de las computadoras ($)          'Costo energía($ / panel)

CostoEnergiaXP143 = (((((WBombillas * NumBombillas * 150) / 1000) * KwEnergia) + (((WAC * 150) / 1000) *
KwEnergia) + (((WXP143 * 150) / 1000) * KwEnergia) + (((WCompresor * 150) / 1000) * KwEnergia) +
(((WComputadoras * NumComputadoras * 150) / 1000) * KwEnergia)) / (PanelesDiario * NumEstaciones))

CostoTotalXP143 = CostosXP143 + CostoEnergiaXP143

'Cálculo de costos de la XP143 Real(estación 2)

'Alistamiento ($/panel)          'Componentes ($/panel) (el programa ya me da el total del costo
multiplicado por el num de componentes que se usan por panel 'Panel ($/panel) Área

CostosXP143Real = ((HrSetupXP143 / PanelesDiario) * MdeOD) + (SumCompCost) + (MdeOD /
(ComponentesChargerXP143 * CambioRolloXP143 * PanelesDiario)) + ((AreaEdificio * Edificio) / (PanelesDiario
* NumEstaciones))

'Consumo de energía de las bombillas ($)          'Consumo de energía del A/C ($)
'Consumo de energía de la XP143 ($)          'Consumo de energía del compresor ($)          "Consumo de energía
de las computadoras ($)          'Costo energía($ / panel)

CostoEnergiaXP143Real = (((((WBombillas * NumBombillas * TiempoXP143) / 1000) * KwEnergia) + (((WAC *
TimeForProdMin) / 1000) * KwEnergia) + (((WXP143 * TimeForProdMin) / 1000) * KwEnergia) +
(((WCompresor * TimeForProdMin) / 1000) * KwEnergia) + (((WComputadoras * NumComputadoras *
TimeForProdMin) / 1000) * KwEnergia)) / (PanelesDiario * NumEstaciones))

CostoTotalXP143Real = CostosXP143Real + CostoEnergiaXP143Real

```

```

'pc ess
'costosactuales = "C:\Users\Enrique\Desktop\eventdata may4\Mach\eventdata\ip3\" & ProductIDNum & ".txt"
'pc fujiflexa
'costosactuales = "C:\FujiFlexa\Server\Data\Prod\ip3rum\" & ProductIDNum & ".txt"
'/////////////////////////////////////////////////////////////////

DataRec.sta(3, 2) = SumCompCost2
DataRec.sta(4, 2) = ErrorCount
DataRec.sta(5, 2) = CostoTotalXP143 + SumCompCost
'volver a escribir el record quitando la señal para liberarlo y poniendo los datos
DataRec.sta(3, 0) = 0
Seek #3, recptr
Put #3, , DataRec
'guardar en el record de control (300) el pointer del ultimo record escrito
Seek #3, 300
Get #3, , DataRec
'***$$***
DataRec.sta(3, 0) = recptr
'***
Seek #3, 300
Put #3, , DataRec
'prepararnos para la próxima posición en el random file
recptr = recptr + 1
Return
'-----

CheckControl:
'Chctrl = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\febrero 2011\Control v-6.exe"
'Abrir el file Control
'Open Chctrl For Random As #8
Seek #3, 1
Get #3, , DataRec
'si es 3, Control nos indica que acabemos el programa
'***$$***
If DataRec.sta(0, 2) = 3 Then
Close
End
End If
'si no es 0 ni 3, será 1, indicación de que Control quiere

```

'que entremos en pausa. Poner 2 para indicar a Control que

'hemos entrado en pausa.

\$\$\$

If DataRec.sta(0, 2) = 1 Then

 DataRec.sta(0, 2) = 2

 Seek #3, 1

 Put #3, , DataRec

 If fsize = 0 Then

 xx = 3

 Module6.Estaciones

 End

 End If

End If

 \$\$\$

If DataRec.sta(0, 2) >= 1 Then GoTo CheckControl

Return

'If fsize >= 1 Then GoTo denuevo

End Sub

Programa “station 3.exe”

Private Type t_DataRec

 sta(5, 4) As Single

End Type

Sub Form_Load()

 Dim DataRec As t_DataRec

 Dim nada As String

 Dim x As String * 1 'se declara de 1 sólo caracter, para poder leer el

 'archivo caracter a caracter

 Dim LineaDeCaracteres As String 'variable donde se irá construyendo la línea

 'de caracteres que se leera uno a uno del

 'archivo.

 Dim filename As String 'nombre del archivo a procesar (eventdata)

 Dim filename1 As String 'nombre del archivo a procesar (stacomb)

 Dim fsize As Long 'tamaño de ese archivo

 Dim fpos As Long 'posición del ultimo caracter leído

 Dim DateTimeStamp As String

 Dim DateStamp As String


```

Dim DateStampSav As String
Dim TimeStamp As String
Dim TimeStampSav As String
Dim ProductID As String
Dim ProductIDSav As String
Dim TimeForProdMin As Single
Dim ProductSlot As Integer
Dim ProdSlotTable(10, 60, 2) As Single
Dim ProductSlot2 As Integer
Dim ProdSlotTable2(10, 60, 2) As Single
Dim ProdSlotID(10) As Long
Dim ProdSlotID2(10) As Long
Dim ErrorCount As Integer
Dim CompNum As Integer
Dim ExtraComp As Integer
Dim SlotNum As Integer
Dim PrimeraLinea As String
Dim Linea As String
Dim recptr As Integer

```

```

'=====
STATION 3

```

```

'=====

```

```

DoEvents

```

```

'-----

```

```

nada = ""

```

```

recptr = 1

```

```

swPrimeraLinea = 0

```

```

PrimeraLinea = ""

```

```

fsize = 0

```

```

swProdStart = 0

```

```

On Error Resume Next

```

```

'esta sección se encarga de cotejar si el archivo a leer tiene datos

```

```

'en él para procesar, usando como criterio su tamaño. Se crea un loop que

```

```

'se repetira, gastando un segundo en cada vuelta

```

```

'hasta que aparezca que hay datos en el mismo.

```

```

'cp ESS

```

```

filename = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\eventdata2.txt"

```

```

'Fuji Flexa cp
'filename = "C:\FujiFlexa\Server\Data\Prod\Xp143\eventdata.txt"
getsize:
fsize = FileLen(filename)
'si el if NO se cumple (hay datos), se sigue abajo y pasamos a abrir
'el archivo
If fsize < 1 Then
For er = 1 To 100
GoSub Waste1      'ir a perder un segundo, para dar tiempo a que
                  'el archivo aparezca con datos
Next er
End If
'GoTo getsize
'se abre el archivo de texto como si fuera un random file, del cual
'se van a leer records de UN caracter de largo
Open filename For Random As #1 Len = 1
fpos = Seek(1) 'fpos se usa para guardar la posición del caracter
          'que se va a leer del archivo
'cp ESS
filename1 = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\stacombnew.dat"
'Fuji Flexa pc
'filename1 = "C:\FujiFlexa\Server\Data\Prod\IP3RUM\stacombnew.dat"
Open filename1 For Random As #3 Len = Len(DataRec)
'el random file que acabamos de abrir, debe haberse creado con otro programa
'que crea un archivo de 300 records. Las primeras cinco posiciones (20 bytes)
'del record 1 y del record 300 son datos de control. Después siguen 5 grupos
'de 5 posiciones cada una, un grupo para cada estación. La primera posición
'de cada grupo se utilizará para record locking, las otras cuatro para
'escribir los datos de salida: tiempo, componentes, errores y comp. extras
'Los datos de control en el record 1 son para indicar a los programas
'que procesen, pausen o dejen de ejecutar. Los datos de control del
'record 300 son para ir registrando el último record que han escrito cada
'programa de estaciones.
'-----
replloop = 1
SumCompCost = 0
SumExtraComp = 0

```

```

ErrorCount = 0
While (reloop)
'fsize = 0 'nos preparamos para el error file no found que puede darse
    'más abajo cuando estemos procesando los eof
swbad = 0    'switch que indica caracter malo leído(p.e. end of file)
LineaDeCaracteres = "" 'variable donde se construirá la línea de caracteres
    'leída. La limpiamos en cada vuelta
'aquí empieza el loop de leer los caracteres de una línea
rep = 1      'esta variable mantiene el while dando ciclos si es uno, o
    'lo acaba si es cero (true or false)
While (rep)  'loop que se repetirá hasta que hayamos leído todos los
    'caracteres que componen una línea del archivo.
getit:
Get #1, , x
If Asc(x) = 10 Then GoTo getit 'si es un carácter line feed (chr(10)) lo
    'ignoramos y leemos otro
Debug.Print x;
fpos = Seek(1)
swbad = 0      'limpiamos este indicador de carácter malo
'Mirar si podemos continuar procesado
GoSub CheckControl
'en cuanto el flag sea 0, hemos salido de pausa, por lo que seguiremos adelante
'-----
'Atrapar los caracteres que puedan indicar fin de archivo (eof).
'Esto se puede dar en medio del archivo mientras se está escribiendo
'o al final del archivo, cuando ya está cerrado
If Asc(x) = 26 Or Asc(x) = 0 Then
    Seek #1, fpos - 1    'volver a posicionar el archivo en el carácter malo
    GoSub Waste1        'esperar un segundo (para dar oportunidad de que se
        'escriba un carácter bueno
    If (swPrimeraLinea) Then
denuevo:
        Close 1
        GoSub CheckControl
        fsize = 0
        fsize = FileLen(filename)
        If fsize < 1 Then GoTo denuevo 'si ha habido error file not found

```

```

'reintentar abrirlo
Open filename For Input As #1
Line Input #1, Linea
If EOF(1) Then GoTo denuevo 'la linea no estaba completa, reintentar
Close 1
Open filename For Random As #1 Len = 1
If Linea <> PrimeraLinea Then
PrimeraLinea = Linea
fpos = 1 'si las lineas no son iguales es que es un archivo nuevo, y
'por tanto leo desde el principio (fpos=1). Si no es archivo
'nuevo me quedo con el fpos que traía de la lectura del archivo y
'sigo desde ahí.
End If
Seek #1, fpos - 1 'tengo que volver a posicionar el archivo en el caracter
'malo, porque si pasé por el line input me lo varió
End If
swbad = 1 'marcar que hemos leído un caracter malo, para no añadirlo
'a la línea que estamos construyendo
End If
'-----
'el bloque de if que sigue se ejecutara si no es un caracter malo.
'Aquí es donde se construye la línea con los caracteres que estamos leyendo
If swbad = 0 Then
'si el caracter leído es chr(13) [RETURN, indica fin de línea], no añadirlo
'a la línea y salir para procesar la línea completa
If x = Chr(13) Then rep = 0 Else LineaDeCaracteres = LineaDeCaracteres & x
End If
Wend 'aqui acaba el ciclo de lectura caracter a caracter, para una linea
'-----
' Sección de procesar la línea leída
'cotejar si se ha guardado la primerísima linea leída
'si no, guardarla y marcar que se tiene guardada
If swPrimeraLinea = 0 Then
PrimeraLinea = LineaDeCaracteres
swPrimeraLinea = 1
End If
'procesar la línea de caracteres

```

```

If InStr(1, LineaDeCaracteres, "PRODSTART") Then GoSub ProdStart
If InStr(1, LineaDeCaracteres, "PDCOUNT") Then GoSub PdCount
If InStr(1, LineaDeCaracteres, "ERROR") Then GoSub ProError
Wend      'wend de todo el proceso, que incluye reabrir el archivo
fin:
Close
End
'subrutina que gasta (waste) un segundo
Waste1:
Debug.Print Estacion2
GoTo denuevo
End
'=====

ProdStart:
'extraer la fecha
P1 = InStr(1, LineaDeCaracteres, ":")
P2 = InStr(P1 + 1, LineaDeCaracteres, ":")
DateTimeStamp = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)
DateStamp = Left(DateTimeStamp, 8)
TimeStamp = Right(DateTimeStamp, 8)
'darle formato normal a la fecha
DateStamp = Mid(DateStamp, 5, 2) & "/" & Right(DateStamp, 2) & "/" & Left(DateStamp, 4)
'extraer el product ID completo y el número solamente (6 digitos)
P1 = InStr(1, LineaDeCaracteres, Chr(9))
P1 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
P1 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
P2 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
ProductID = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)
ProductIDNum = Left(ProductID, 6)
'Como el tiempo total para producir un producto se toma desde el inicio de
'un producto hasta el inicio del próximo, la primera vez que se procesa
'un PROSTART no hacemos cálculo de tiempo. Guardamos los valores obtenidos
'hasta el próximo PRODSTART.
If swProdStart = 0 Then
    TimeStampSav = TimeStamp
    DateStampSav = DateStamp
End If

```

```

'Las veces subsiguientes, el encontrar un
'PRODSTART (entrar en esta subrutina) señala que debemos hacer el cálculo
'y producir una línea de resultados, que es lo que se hace en el siguiente
'bloque de IF.
If swProdStart = 1 Then
difh = Left(TimeStamp, 2) - Left(TimeStampSav, 2)
difm = Mid(TimeStamp, 3, 2) - Mid(TimeStampSav, 3, 2)
difs = Mid(TimeStamp, 5, 2) - Mid(TimeStampSav, 5, 2)
difmili = Right(TimeStamp, 2) - Right(TimeStampSav, 2)
TimeForProdMin = difh * 60 + difm + difs / 60 + difmili / 6000
GoSub DataWrite
'If nada = "" Then nada = InputBox("ENTER para seguir")
Print #4, TimeForProdMin * 2, TimeForProdMin, ProductID, SumCompCost
SumCompCost = 0
SumExtraComp = 0
ErrorCount = 0
TimeStampSav = TimeStamp
End If
swProdStart = 1
Return
'-----
ProError:
If swProdStart = 0 Then Return
ErrorCount = ErrorCount + 1
Return
'-----
'-----
PdCount:
Debug.Print "pdcount"
If swProdStart = 0 Then Return
'extraer el numero de componentes y de esos, los extras que hubo que usar
P1 = InStr(1, LineaDeCaracteres, Chr(9))
For k = 1 To 5
P1 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
Next k
P2 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
ProductSlot = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)

```

```

GoSub GetSlotCost
P1 = P2
P1 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
P2 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
CompNum = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)
P1 = P2
P2 = InStr(P1 + 1, LineaDeCaracteres, Chr(9))
ExtraComp = Mid(LineaDeCaracteres, P1 + 1, P2 - P1 - 1)
SumExtraComp = SumExtraComp + ExtraComp
SumCompCost = SumCompCost + CompNum * SlotCost
SumCompCost2 = SumCompCost + SumExtraComp * SlotCost
Debug.Print SumCompCost
Debug.Print SumCompCost2
Return
'-----

GetSlotCost:
'Buscar el product ID Num en la tabla
swfound = 0
For i = 1 To 10
  If ProdSlotID(i) = ProductIDNum Then
    swfound = i
    GoTo GetCost
  End If
Next i
'pc ess
slotsfilename = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\" & ProductIDNum & ".txt"
'pc fujiflexa
'slotsfilename = "C:\FujiFlexa\Server\Data\Prod\ip3rum\" & ProductIDNum & ".txt"
Open slotsfilename For Input As #2
'Buscar dimensión vacia en el array
i = 1
While (ProdSlotID(i))
  i = i + 1
Wend
'Cargar los datos de la tabla en cuestion
ProdSlotID(i) = ProductIDNum
j = 0

```

```

While Not EOF(2)
    j = j + 1
    Input #2, ProdSlotTable(i, j, 1)
    Input #2, dummy
    Input #2, ProdSlotTable(i, j, 2)
Wend
SlotLines = j
GetCost:
'seleccionar el costo del slot
SlotCost = 0
j = 1
While (ProdSlotTable(i, j, 1))
If ProductSlot = ProdSlotTable(i, j, 1) Then
SlotCost = ProdSlotTable(i, j, 2)
Close 2
Return
End If
j = j + 1
Wend
Close 2
Return
'-----
DataWrite:
'Aqui se implementa record locking que dada la naturaleza del proceso
'(pasa un tiempo entre que una estación escriba y escriba la siguiente)
'me parece innecesario. Pero se deja como un ejercicio de programación
'y por si alguna vez se modifica los programas para hacer cosas con
'el archivo donde haya riesgo que un programa escriba datos a la vez que
'otro actualiza.
'primero asegurar que no hay otro programa escribiendo
'en el record, o que estamos en pausa.
Seek #3, recptr
Get #3, , DataRec
tlock = 0
For il = 1 To 5
tlock = tlock + DataRec.sta(il, 0)
Next il

```



```

'si la suma (tlock) no es cero, hay alguien usando el record
'volver a leerlo hasta que quede libre
If tlock < 0 Then GoTo DataWrite
'poner la señal de que queremos usar el record
DataRec.sta(2, 0) = 1
Seek #3, recptr
Put #3, , DataRec
Seek #3, recptr
Get #3, , DataRec
'*****3 por 1
'Columna 11 a 14 porque son las 3eras 5 columnas
'DataRec.sta(2, 1) = TimeForProdMin
DataRec.sta(2, 2) = SumCompCost + CostoTotalIPIII
'\////////////////////////////////////
'=====
'=====

costos:
'Cálculo de costos de la IPIII (estación 3)

'Alistamiento ($/panel)          'Componentes ($/panel) (el programa ya me da el total del costo
multiplicado por el num de componentes que se usan por panel          'Panel ($/panel)

CostosIPIII = ((HrSetupIPIII / PanelesDiario) * MdeOD) + (CostosComponentes) + (MdeOD /
(ComponentesChargerXP143 * CambioRolloXP143 * PanelesDiario)) + ((AreaEdificio * Edificio) / (PanelesDiario
* NumEstaciones))

'Consumo de energía de las bombillas ($)          'Consumo de energía del A/C ($)
'Consumo de energía de la IPIII ($)          'Consumo de energía del compresor ($)          "Consumo de energía de
las computadoras ($)          'Costo energía($ / panel)

CostoEnergiaIPIII = (((((WBombillas * NumBombillas * 155) / 1000) * KwEnergia) + (((WAC * 155) / 1000) *
KwEnergia) + (((WIPIII * 155) / 1000) * KwEnergia) + (((WCompresor * 155) / 1000) * KwEnergia) +
(((WComputadoras * NumComputadoras * 155) / 1000) * KwEnergia)) / (PanelesDiario * NumEstaciones))

CostoTotalIPIII = CostosIPIII + CostoEnergiaIPIII

'Cálculo de costos de la IPIII Real(estación 3)

'Alistamiento ($/panel)          'Componentes ($/panel) (el programa ya me da el total del costo
multiplicado por el num de componentes que se usan por panel          'Panel ($/panel)

CostosIPIIIReal = ((HrSetupIPIII / PanelesDiario) * MdeOD) + (CostosComponentes) + (MdeOD /
(ComponentesChargerXP143 * CambioRolloXP143 * PanelesDiario)) + ((AreaEdificio * Edificio) / (PanelesDiario
* NumEstaciones))

```

```

'Consumo de energía de las bombillas ($)          'Consumo de energía del A/C ($)
'Consumo de energía de la IPIII ($)          'Consumo de energía del compresor ($)          "Consumo de energía de
las computadoras ($)          'Costo energía($ / panel)

```

```

CostoEnergiaIPIIIReal = (((((WBombillas * NumBombillas * TimeForProdMin) / 1000) * KwEnergia) + (((WAC *
TimeForProdMin) / 1000) * KwEnergia) + (((WIPIII * TimeForProdMin) / 1000) * KwEnergia) + (((WCompresor
* TimeForProdMin) / 1000) * KwEnergia) + (((WComputadoras * NumComputadoras * TimeForProdMin) / 1000)
* KwEnergia)) / (PanelesDiario * NumEstaciones))

```

```

CostoTotalIPIIIReal = CostosIPIIIReal + CostoEnergiaIPIIIReal

```

```

'/////////////////////////////////////////////////////////////////

```

```

DataRec.sta(3, 2) = CostoTotalIPIIIReal + SumCompCost2

```

```

DataRec.sta(4, 2) = ErrorCount

```

```

DataRec.sta(5, 2) = SumExtraComp

```

```

'volver a escribir el record quitando la señal para liberarlo y poniendo los datos

```

```

DataRec.sta(3, 0) = 0

```

```

Seek #3, recptr

```

```

Put #3, , DataRec

```

```

'guardar en el record de control (300) el pointer del ultimo record escrito

```

```

Seek #3, 300

```

```

Get #3, , DataRec

```

```

'***$$***

```

```

DataRec.sta(3, 0) = recptr

```

```

Seek #3, 300

```

```

Put #3, , DataRec

```

```

'prepararnos para la próxima posición en el random file

```

```

recptr = recptr + 1

```

```

Return

```

```

'-----

```

```

CheckControl:

```

```

'Chctrl = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\febrero 2011\Control v-6.exe"

```

```

'Abrir el file Control

```

```

'Open Chctrl For Random As #8

```

```

Seek #3, 1

```

```

Get #3, , DataRec

```

```

'si es 3, Control nos indica que acabemos el programa

```

```

'***$$***

```

```

If DataRec.sta(0, 2) = 3 Then

```

```

Close

```

```

End
End If
'si no es 0 ni 3, será 1, indicación de que Control quiere
'que entremos en pausa. Poner 2 para indicar a Control que
'hemos entrado en pausa.
***$$$***
If DataRec.sta(0, 2) = 1 Then
    DataRec.sta(0, 2) = 2
    Seek #3, 1
    Put #3, , DataRec
    If fsize = 0 Then
        xx = 3
        Module6.Estaciones
    End
End If
End If
***$$$***
If DataRec.sta(0, 2) >= 1 Then GoTo CheckControl
Return
'If fsize >= 1 Then GoTo denuevo
End Sub

```

Apéndice 6 Módulo “Control.exe”

El archivo “Control.exe”, controla el funcionamiento principal del programa de este proyecto. Consiste en un código de VB6®, que tiene cuatro botones. Uno rojo, que es de **STOP**, otro amarillo que es de **PAUSE**, otro verde, que es de **RUN** y otro gris que es de **EXIT**. Esos botones obligan a los programas de cada estación, a que hagan lo que les indica el botón correspondiente. En el control, también hay dos líneas de datos, la línea superior indica los números de la situación de cada una de las cinco estaciones (0=RUN; 1=PAUSE; 2=PAUSE (máquina responde al control); 3=STOP); y la segunda línea, indica los últimos renglones en los que se ha puesto información en el archivo “stacomb.dat” de cada estación.

	STAT 1	STAT 2	STAT 3	STAT 4	STAT 5
Status ST's	0	0	0	0	0
# de Record	0	94	0	0	0

STOP PAUSE RUN EXIT

Código del módulo “control.exe”:

```

Type t_DataRec
sta(5, 4) As Single
End Type
Public DataRec As t_DataRec

Sub Main()
Project1.Form1.Visible = True
DoEvents
Call w
End Sub

Sub w()
While (1)
Seek #3, 300
Get #3, , DataRec
Form1.Rec_Stat1.Caption = DataRec.sta(0, 0)
Form1.Rec_Stat2.Caption = DataRec.sta(0, 1)
Form1.Rec_Stat3.Caption = DataRec.sta(0, 2)
Form1.Rec_Stat4.Caption = DataRec.sta(0, 3)
Form1.Rec_Stat5.Caption = DataRec.sta(0, 4)
Seek #3, 1
Get #3, , DataRec
Form1.STA1_Status.Caption = DataRec.sta(0, 0)
Form1.STA2_Status.Caption = DataRec.sta(0, 1)
Form1.STA3_Status.Caption = DataRec.sta(0, 2)
Form1.STA4_Status.Caption = DataRec.sta(0, 3)
Form1.STA5_Status.Caption = DataRec.sta(0, 4)
DoEvents
Wend
End Sub

```

'El comando EXIT cierra todos los files.

```

Private Sub cmdExit_Click()
Close
End
End Sub

```

'El comando PAUSE anota en las celdas correspondientes a la estación en cuestión,
'un "1" que eso significa la orden PAUSE
'Rec_stat1 significa = el numero de record de la estacion 1

```

Private Sub cmdPause_Click()
Seek #3, 1
Get #3, , DataRec
DataRec.sta(0, 0) = 1
DataRec.sta(0, 1) = 1
DataRec.sta(0, 2) = 1
DataRec.sta(0, 3) = 1
DataRec.sta(0, 4) = 1
Seek #3, 1
Put #3, , DataRec

```

```
'pausar las estaciones
Form1.STA1_Status.Caption = DataRec.sta(0, 0)
Form1.STA2_Status.Caption = DataRec.sta(0, 1)
Form1.STA3_Status.Caption = DataRec.sta(0, 2)
Form1.STA4_Status.Caption = DataRec.sta(0, 3)
Form1.STA5_Status.Caption = DataRec.sta(0, 4)
End Sub
```

'El comando RUN anota en las celdas correspondientes a la estación en cuestión,
'un "0" que eso significa la orden RUN

```
Private Sub cmdRun_Click()
Seek #3, 1
Get #3, , DataRec
DataRec.sta(0, 0) = 0
DataRec.sta(0, 1) = 0
DataRec.sta(0, 2) = 0
DataRec.sta(0, 3) = 0
DataRec.sta(0, 4) = 0
Seek #3, 1
Put #3, , DataRec
'arrancar stations
Form1.STA1_Status.Caption = DataRec.sta(0, 0)
Form1.STA2_Status.Caption = DataRec.sta(0, 1)
Form1.STA3_Status.Caption = DataRec.sta(0, 2)
Form1.STA4_Status.Caption = DataRec.sta(0, 3)
Form1.STA5_Status.Caption = DataRec.sta(0, 4)
End Sub
```

'El comando STOP anota en las celdas correspondientes a la estación en cuestión,
'un "3" que eso significa la orden STOP

```
Private Sub cmdStop_Click()
Seek #3, 1
Get #3, , DataRec
DataRec.sta(0, 0) = 3
DataRec.sta(0, 1) = 3
DataRec.sta(0, 2) = 3
DataRec.sta(0, 3) = 3
DataRec.sta(0, 4) = 3
Seek #3, 1
Put #3, , DataRec
'parar las estaciones
Form1.STA1_Status.Caption = DataRec.sta(0, 0)
Form1.STA2_Status.Caption = DataRec.sta(0, 1)
Form1.STA3_Status.Caption = DataRec.sta(0, 2)
Form1.STA4_Status.Caption = DataRec.sta(0, 3)
Form1.STA5_Status.Caption = DataRec.sta(0, 4)
End Sub
Private Sub Form_Load()
DoEvents
'cp ess
FileName = "C:\Users\Enrique\Desktop\eventdata may4\Mach\eventdata\ip3\stacomb24.dat"
'cp fujiflexa
'FileName = "c:\FujiFlexa\Server\Data\Prod\IP3RUM\stacomb18.dat"
```

```
Open FileName For Random As #3 Len = Len(DataRec)
Seek #3, 1
Get #3, , DataRec
DataRec.sta(0, 0) = 0
DataRec.sta(0, 1) = 0
DataRec.sta(0, 2) = 0
DataRec.sta(0, 3) = 0
DataRec.sta(0, 4) = 0
Seek #3, 1
Put #3, , DataRec
'arrancar stations
Form1.STA1_Status.Caption = DataRec.sta(0, 0)
Form1.STA2_Status.Caption = DataRec.sta(0, 1)
Form1.STA3_Status.Caption = DataRec.sta(0, 2)
Form1.STA4_Status.Caption = DataRec.sta(0, 3)
Form1.STA5_Status.Caption = DataRec.sta(0, 4)
End Sub
```

Apéndice 7 Módulo de VBA® para llenar el archivo "stacomb.dat" de la primera estación

```
Type t_DataRec
sta(5, 4) As Single
End Type
```

```
'=====
'                                     ESTACION 1 nueva version
'=====
```

```
Sub Form_Load()
```

```
xx = 1
```

```
'=====
```

```
Dim DataRec As t_DataRec
```

```
Dim vueltas As Integer
```

```
Dim filename1 As String      'nombre del archivo a procesar (stacomb)
```

```
Dim Time1Stamp As Variant
```

```
Dim Time2Stamp As Variant
```

```
Dim Time11Stamp As Variant
```

```
Dim Time22Stamp As Variant
```

```
Dim TimeForProdMin As Variant
```

```
DoEvents
```

```
'-----
```

```
recptr = 1
```

```
Sheets("Sheet1").Select
```

```
Cells.Select
```

```
Selection.ClearContents
```

```
filename1 = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\stacombnew.dat"
```

```
Open filename1 For Random As #3 Len = Len(DataRec)
```

```
'el random file que acabamos de abrir, debe haberse creado con otro programa
```

```
'que crea un archivo de 300 records. Las primeras cinco posiciones (20 bytes)
```

```
'del record 1 y del record 300 son datos de control. Después siguen 5 grupos
```

```
'de 5 posiciones cada una, un grupo para cada estación. La primera posición
```

```
'de cada grupo se utilizará para record locking, las otras cuatro para
```

```
'escribir los datos de salida: tiempo, componentes, errores y comp. extras
```

```
'Los datos de control en el record 1 son para indicar a los programas
```

```
'que procesen, pausen o dejen de ejecutar. Los datos de control del
```

```
'record 300 son para ir registrando el último record que han escrito cada
```

```
'programa de estaciones.
```

```
Sheet1.Cells(1, 1) = "Numero"
```

```
Sheet1.Cells(1, 2) = "Sensor In"
```

```
Sheet1.Cells(1, 3) = "Sensor Out"
```

```
Sheet1.Cells(1, 4) = "Tiempo S-In"
```

```
Sheet1.Cells(1, 5) = "Tiempo S-Out"
```

```
Sheet1.Cells(1, 6) = "Tiempo estacion"
```

```
Sheet1.Cells(1, 7) = "Costo total DEK"
```

```
Sheet1.Cells(1, 8) = "Costo total DEKReal"
```

```
'si el sensor de entrada de la DEK cambia de estado, entonces se guarda la hora en la que cambio de estado
```

```
'en la celda sta(1,0)
```

```
'Anota el tiempo de entrada del board a la estación.
```


empieza:

aqui = 2

repetir:

'Anota la hora en que el sensor C detecta que hay una pieza pasando

Do Until Sheet1.Cells(aqui, 2) = 1

mychannel = DDEInitiate("datahub", "default")

Application.Worksheets("Sheet1").Activate

InDEK = DDERequest(mychannel, "Channel1.PLC.Salida5")

Sheet1.Cells(aqui, 2) = InDEK

DDETerminate mychannel

vueltas = vueltas + 1

If vueltas = 150 Then

GoTo Final

End If

Loop

Time1Stamp = Format(Time, "hh:mm:ss")

Sheet1.Cells(aqui, 4) = Time1Stamp

InDEK = 0

'Anota la hora en que el sensor D detecta que hay una pieza pasando

Do Until Sheet1.Cells(aqui, 3) = 1

mychannel = DDEInitiate("datahub", "default")

Application.Worksheets("Sheet1").Activate

OutDEK = DDERequest(mychannel, "Channel1.PLC.Salida6")

Sheet1.Cells(aqui, 3) = OutDEK

DDETerminate mychannel

vueltas = vueltas + 1

If vueltas = 150 Then

GoTo Final

End If

Loop

Time2Stamp = Format(Time, "hh:mm:ss")

Sheet1.Cells(aqui, 5) = Time2Stamp

OutDEK = 0

If Sheet1.Cells(aqui, 4) <> Sheet1.Cells(aqui, 5) Then

TimeForProdMin = ((Sheet1.Cells(aqui, 5)) - Int(Sheet1.Cells(aqui, 5))) * 24 - ((Sheet1.Cells(aqui, 4)) - Int(Sheet1.Cells(aqui, 4))) * 24

Sheet1.Cells(aqui, 6) = TimeForProdMin

End If

'=====

'Cálculo de costos de la DEK TEORICO (estación 1)

'Manejo de materiales (\$/panel)

'Mano de obra Ingeniero (\$/paneles)

'Alistamiento (\$/panel)

'Plomo (\$/panel)

'Panel (\$/panel)

CostoRawMDEK = (MH * ComponentesCharger * ImagenesBoard) + (MdeOIngeniero * HorasIngeniero / PanelesDiario) + ((HrSetupDEK / PanelesDiario) * MdeOD) + (BotePlomo / BoardsPlomo) + (Panel / CantidadBoards) + ((AreaEdificio * Edificio) / (PanelesDiario * NumEstaciones))

'Consumo de energía de las bombillas (\$)

'Consumo de energía del A/C (\$)

'Consumo de energía del A/C (\$)

'Consumo de energía del compresor (\$)

'Consumo de energía de las

computadoras (\$)

'Costo energía(\$ / panel)

CostoEnergiaDEK = (((WBombillas * NumBombillas * 12.5) / 1000) * KwEnergia) + (((WAC * 12.5) / 1000) * KwEnergia) + (((WDEK * 12.5) / 1000) * KwEnergia) + (((WCompresor * 12.5) / 1000) * KwEnergia) + (((WComputadoras * NumComputadoras * 12.5) / 1000) * KwEnergia) / (PanelesDiario * NumEstaciones))

CostoTotalDEK = CostoRawMDEK + CostoEnergiaDEK

Sheet1.Cells(aquí, 7) = CostoTotalDEK 'Anotar las ecuaciones de costo teorico

'Cálculo de costos de la DEK REAL (estación 1)

'Manejo de materiales (\$/panel) 'Mano de obra Ingeniero (\$/paneles)
 'Alistamiento (\$/panel) 'Plomo (\$/panel) 'Panel (\$/panel)
 CostoRawMDEKReal = (MH * ComponentesCharger * ImagenesBoard) + (MdeOIngeniero * HorasIngeniero /
 PanelesDiario) + ((HrSetupDEK / PanelesDiario) * MdeOD) + (BotePlomo / BoardsPlomo) + (Panel /
 CantidadBoards) + ((AreaEdificio * Edificio) / (PanelesDiario * NumEstaciones))

'Consumo de energía de las bombillas (\$) 'Consumo de energía del A/C (\$)
 'Consumo de energía del A/C (\$) 'Consumo de energía del compresor (\$) "Consumo de energía de las
 computadoras (\$) 'Costo energía(\$ / panel)
 CostoEnergiaDEKReal = (((((WBombillas * NumBombillas * TimeForProdMin) / 1000) * KwEnergia) + (((WAC *
 TimeForProdMin) / 1000) * KwEnergia) + (((WDEK * TimeForProdMin) / 1000) * KwEnergia) + (((WCompresor
 * TimeForProdMin) / 1000) * KwEnergia) + (((WComputadoras * NumComputadoras * TimeForProdMin) / 1000)
 * KwEnergia)) / (PanelesDiario * NumEstaciones))

CostoTotalDEKReal = CostoRawMDEKReal + CostoEnergiaDEKReal

Sheet1.Cells(aquí, 8) = CostoTotalDEKReal ' anotar la ecuacion de costo real

'=====

aquí = aquí + 1

GoTo repetir

'-----

DataWrite:

'Aquí se implementa record locking que dada la naturaleza del proceso
 '(pasa un tiempo entre que una estación escriba y escriba la siguiente)
 'me parece innecesario. Pero se deja como un ejercicio de programación
 'y por si alguna vez se modifica los programas para hacer cosas con
 'el archivo donde haya riesgo que un programa escriba datos a la vez que
 'otro actualiza.

'primero asegurar que no hay otro programa escribiendo
 'en el record, o que estamos en pausa.
 Seek #3, recptr

Get #3, , DataRec

tlock = 0

' For il = 1 To 5

' tlock = tlock + DataRec.sta(il, 0)

' Next il

'si la suma (tlock) no es cero, hay alguien usando el record
 'volver a leerlo hasta que quede libre
 If tlock > 0 Then GoTo DataWrite

'poner la señal de que queremos usar el record

DataRec.sta(1, 0) = 1

Seek #3, recptr

Put #3, , DataRec

Seek #3, recptr

Get #3, , DataRec

```

DataRec.sta(1, 1) = CostoTotalDEK
DataRec.sta(1, 2) = CostoTotalDEKReal
Debug.Print recptr, DataRec.sta(1, 0), DataRec.sta(1, 1), DataRec.sta(1, 2)

'volver a escribir el record quitando la señal para liberarlo y poniendo los datos
DataRec.sta(0, 1) = 0
Seek #3, recptr
Put #3, , DataRec
'guardar en el record de control (300) el pointer del ultimo record escrito
Seek #3, 300
Get #3, , DataRec
'***$$***
DataRec.sta(0, 0) = recptr
'***
'

Seek #3, 300
Put #3, , DataRec
'prepararnos para la próxima posición en el random file
recptr = recptr + 1
'-----
CheckControl:

'Chctrl = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\febrero 2011\Control final.exe"
'Abir el file Control
'Open Chctrl For Random As #8
'Close #8
Seek #3, 1
Get #3, , DataRec
'si es 3, Control nos indica que acabemos el programa
'***$$***
If DataRec.sta(0, 0) = 3 Then
Close
End
End If
'si no es 0 ni 3, será 1, indicación de que Control quiere
'que entremos en pausa. Poner 2 para indicar a Control que
'hemos entrado en pausa.
'***$$***
If DataRec.sta(0, 0) = 1 Then
DataRec.sta(0, 0) = 2
Seek #3, 1
Put #3, , DataRec
End If
'***$$***
If DataRec.sta(0, 0) >= 1 Then End
Final:

Close #3
xx = xx + 1
' Call Module8.GenerarDataChart 'Graficar estacion 2
Module6.Estaciones

End Sub

```

Apéndice 8 Módulo de VB6® para llenar el archivo "stacomb.dat" de la segunda estación

```
Type t_DataRec
```

```
sta(5, 4) As Single
```

```
End Type
```

```
'Generar estacion 2
```

```
Sub GenerarDataChart()
```

```
Dim DataRec As t_DataRec
```

```
filename = "C:\Users\Enrique\Desktop\eventdata may4\Mach\eventdata\ip3\stacomb100.dat"
```

```
Open filename For Random As #3 Len = Len(DataRec)
```

```
GoSub CleanSheet
```

```
posVuelta = 1
```

```
For j = 1 To 3
```

```
ptrRec = posVuelta
```

```
For i = 3 To 32
```

```
repetir:
```

```
Seek #3, 1: Get #3, , DataRec
```

```
DoEvents
```

```
If DataRec.sta(0, 0) = 3 Then Close: End
```

```
Seek #3, ptrRec: Get #3, , DataRec
```

```
Debug.Print ptrRec, DataRec.sta(1, 1), DataRec.sta(2, 1), DataRec.sta(3, 1), DataRec.sta(4, 1), DataRec.sta(5, 1)
```

```
DoEvents
```

```
prod = DataRec.sta(1, 1) * DataRec.sta(2, 1) * DataRec.sta(3, 1) * DataRec.sta(4, 1) * DataRec.sta(5, 1)
```

```
If prod <> 0 Then GoTo repetir
```

```
Costos_Reales = DataRec.sta(2, 2)
```

```
Errores = DataRec.sta(2, 3)
```

```
CompExtras = DataRec.sta(2, 4)
```

```
If Errores <> 0 Then ErroresChart = 1.01 * Costos_Reales Else ErroresChart = 0
```

```
If CompExtras <> 0 Then CompExtrasChart = 1.04 * Costos_Reales Else CompExtrasChart = 0
```

```
Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 2) = Costos_Reales
```

```
Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 3) = Errores
```

```
Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 5) = CompExtras
```

```
Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 4) = ErroresChart
```

```
Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 6) = CompExtrasChart
```

```
ptrRec = ptrRec + 1
```

```
DoEvents
```

```
    Next i
For w = 1 To 10
Next w
posVuelta = posVuelta + 1
Next j
Close
End
CleanSheet:
    For i = 3 To 32
        Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 2) = Costos_Reales
        Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 3) = Errores
        Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 5) = CompExtras
        Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 4) = ErroresChart
        Workbooks("estaciones 1-4-5.xlsm").Worksheets("GenerarDataChart").Cells(i, 6) = CompExtrasChart
    Next i
    DoEvents
Return
End Sub
```

Apéndice 9 Módulo de VB6® para llenar el archivo "stacomb.dat" de la tercera estación

```

Type t_DataRec
sta(5, 4) As Single
End Type

'Chart estacion 3
Sub GenerarDataChart3()
Dim DataRec As t_DataRec
filename = "C:\Users\Enrique\Desktop\eventdata may4\Mach\eventdata\ip3\stacomb24.dat"
Open filename For Random As #3 Len = Len(DataRec)
GoSub CleanSheet
posVuelta = 1
For j = 1 To 3
ptrRec = posVuelta
    For i = 3 To 32
repetir:
        Seek #3, 1: Get #3, , DataRec
        DoEvents
        If DataRec.sta(0, 0) = 3 Then Close: End
        Seek #3, ptrRec: Get #3, , DataRec
        Debug.Print ptrRec, DataRec.sta(1, 2), DataRec.sta(2, 2), DataRec.sta(3, 2), DataRec.sta(4, 2), DataRec.sta(5, 2)
        DoEvents
        prod = DataRec.sta(1, 2) * DataRec.sta(2, 2) * DataRec.sta(3, 2) * DataRec.sta(4, 2) * DataRec.sta(5, 2)
        If prod <> 0 Then GoTo repetir
        Costos_Reales = DataRec.sta(3, 2)
        Errores = DataRec.sta(3, 3)
        CompExtras = DataRec.sta(3, 4)
        If Errores <> 0 Then ErroresChart = 1.01 * Costos_Reales Else ErroresChart = 0
        If CompExtras <> 0 Then CompExtrasChart = 1.04 * Costos_Reales Else CompExtrasChart = 0
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 2) = Costos_Reales
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 3) = Errores
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 5) = CompExtras
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 4) = ErroresChart
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 6) = CompExtrasChart
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 7) = Costos_Teoricos
        ptrRec = ptrRec + 1
        DoEvents
    
```

```

    Next i
For w = 1 To 10
Next w
posVuelta = posVuelta + 1
Next j
Close
End
CleanSheet:
    For i = 3 To 32
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 2) = Costos_Reales
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 3) = Errores
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 5) = CompExtras
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 4) = ErroresChart
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 6) = CompExtrasChart
        Workbooks("estaciones 1-4-5 v3.xlsm").Worksheets("GenerarDataChart3").Cells(i, 7) = Costos_Teoricos
    Next i
    DoEvents
Return
End Sub

```

Apéndice 10 Módulo de VBA® para llenar el archivo "stacomb.dat" de la cuarta estación

```
Type t_DataRec
sta(5, 4) As Single
End Type
```

```
'=====
'                                     ESTACION 4 nueva version
'=====
```

```
Sub Form_Load()
```

```
xx = 4
```

```
'=====
Dim DataRec As t_DataRec
Dim vueltas As Integer
Dim filename1 As String      'nombre del archivo a procesar (stacomb)
Dim Time1Stamp As Variant
Dim Time2Stamp As Variant
Dim Time11Stamp As Variant
Dim Time22Stamp As Variant
Dim TimeForProdMin As Variant
'Dim InDEK As Integer
'Dim OutDEK As Integer
'Dim recptr As Integer
```

```
DoEvents
```

```
'-----
recptr = 1
```

```
Sheets("Sheet2").Select
Cells.Select
Selection.ClearContents
filename1 = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\stacombnew.dat"
Open filename1 For Random As #3 Len = Len(DataRec)
'el random file que acabamos de abrir, debe haberse creado con otro programa
'que crea un archivo de 300 records. Las primeras cinco posiciones (20 bytes)
'del record 1 y del record 300 son datos de control. Después siguen 5 grupos
'de 5 posiciones cada una, un grupo para cada estación. La primera posición
'de cada grupo se utilizará para record locking, las otras cuatro para
'escribir los datos de salida: tiempo, componentes, errores y comp. extras
'Los datos de control en el record 1 son para indicar a los programas
'que procesen, pausen o dejen de ejecutar. Los datos de control del
'record 300 son para ir registrando el último record que han escrito cada
'programa de estaciones.
```

```
'-----
'GoSub CheckControl
'-----
```

```
Sheet2.Cells(1, 1) = "Numero"
Sheet2.Cells(1, 2) = "Sensor In"
Sheet2.Cells(1, 3) = "Sensor Out"
Sheet2.Cells(1, 4) = "Tiempo S-In"
```



```

Sheet2.Cells(1, 5) = "Tiempo S-Out"
Sheet2.Cells(1, 6) = "Tiempo estacion"
Sheet2.Cells(1, 7) = "Costo total INSPECCION"
Sheet2.Cells(1, 8) = "Costo total INSPECCIONReal"
'Anota el tiempo de entrada del board a la estación.
empieza:

```

```

aqui = 2
'Sheet2.Cells(2, 1) = 1
'Sheet2.Cells(3, 1) = 2

```

repetir:

```

'Anota la hora en que el sensor D detecta que hay una pieza pasando
Do Until Sheet2.Cells(aqui, 2) = 1
    mychannel = DDEInitiate("datahub", "default")
    Application.Worksheets("Sheet2").Activate
    InDEK = DDERequest(mychannel, "Channel1.PLC.Salida5")
    Sheet2.Cells(aqui, 2) = InDEK
    DDETerminate mychannel
    vueltas = vueltas + 1
    If vueltas = 150 Then
        GoTo Final
    End If

```

Loop

```

    Time1Stamp = Format(Time, "hh:mm:ss")
    Sheet2.Cells(aqui, 4) = Time1Stamp
    InDEK = 0

```

```

'Anota la hora en que el sensor E detecta que hay una pieza pasando
Do Until Sheet2.Cells(aqui, 3) = 1

```

```

    mychannel = DDEInitiate("datahub", "default")
    Application.Worksheets("Sheet2").Activate
    OutDEK = DDERequest(mychannel, "Channel1.PLC.Salida6")
    Sheet2.Cells(aqui, 3) = OutDEK
    DDETerminate mychannel
    vueltas = vueltas + 1
    If vueltas = 150 Then
        GoTo Final
    End If

```

Loop

```

    Time2Stamp = Format(Time, "hh:mm:ss")
    Sheet2.Cells(aqui, 5) = Time2Stamp
    OutDEK = 0

```

```

    If Sheet2.Cells(aqui, 4) <> Sheet2.Cells(aqui, 5) Then

```

```

        TimeForProdMin = ((Sheet2.Cells(aqui, 5)) - Int(Sheet2.Cells(aqui, 5))) * 24 - ((Sheet2.Cells(aqui, 4)) -
Int(Sheet2.Cells(aqui, 4))) * 24

```

```

        Sheet2.Cells(aqui, 6) = TimeForProdMin

```

```

    End If

```

```

'=====

```

```

'Cálculo de costos de la INSPECCION TEORICO (estación 4)

```

```

'Panel ($/panel)

```

```

CostosINSPECCION = (MdeOD * PanelesDiario * 90)

```

```

'Consumo de energía de las bombillas ($)
'Consumo de energía del compresor ($)
energía($ / panel)

```

```

'Consumo de energía del A/C ($)
"Consumo de energía de las computadoras ($)
'Costo
CostoEnergiaINSPECCION = (((((WBombillas * NumBombillas * 90) / 1000) * KwEnergia) + (((WAC * 90) / 1000) * KwEnergia) + (((WCompresor * 90) / 1000) * KwEnergia) + (((WComputadoras * NumComputadoras * 90) / 1000) * KwEnergia)) / (PanelesDiario * NumEstaciones)) + ((AreaEdificio * Edificio) / (PanelesDiario * NumEstaciones))

```

```

CostoTotalINSPECCION = CostosINSPECCION + CostoEnergiaINSPECCION

```

```

Sheet2.Cells(aqui, 7) = CostoTotalINSPECCION 'Anotar las ecuaciones de costo teorico
'Cálculo de costos de la INSPECCION REAL (estación 4)

```

```

'Panel ($/panel)
CostosINSPECCIONReal = (MdeOD * PanelesDiario * TiempoINSPECCION)

```

```

'Consumo de energía de las bombillas ($)
'Consumo de energía del compresor ($)
energía($ / panel)

```

```

'Consumo de energía del A/C ($)
"Consumo de energía de las computadoras ($)
'Costo
CostoEnergiaINSPECCIONReal = (((((WBombillas * NumBombillas * TimeForProdMin) / 1000) * KwEnergia) + (((WAC * TimeForProdMin) / 1000) * KwEnergia) + (((WCompresor * TimeForProdMin) / 1000) * KwEnergia) + (((WComputadoras * NumComputadoras * TimeForProdMin) / 1000) * KwEnergia)) / (PanelesDiario * NumEstaciones)) + ((AreaEdificio * Edificio) / (PanelesDiario * NumEstaciones))

```

```

CostoTotalINSPECCIONReal = CostosINSPECCIONReal + CostoEnergiaINSPECCIONReal

```

```

Sheet2.Cells(aqui, 8) = CostoTotalINSPECCIONReal ' anotar la ecuacion de costo real

```

```

'=====
aqui = aqui + 1
GoTo repetir
'-----

```

```

DataWrite:

```

```

'Aqui se implementa record locking que dada la naturaleza del proceso
'(pasa un tiempo entre que una estación escriba y escriba la siguiente)
'me parece innecesario. Pero se deja como un ejercicio de programación
'y por si alguna vez se modifica los programas para hacer cosas con
'el archivo donde haya riesgo que un programa escriba datos a la vez que
'otro actualiza.

```

```

'primero asegurar que no hay otro programa escribiendo
'en el record, o que estamos en pausa.

```

```

Seek #3, recptr
Get #3, , DataRec
tlock = 0

```

```

' For il = 1 To 5
' tlock = tlock + DataRec.sta(il, 0)
' Next il

```

```

'si la suma (tlock) no es cero, hay alguien usando el record
'volver a leerlo hasta que quede libre

```

```

If tlock > 0 Then GoTo DataWrite

```

```

'poner la señal de que queremos usar el record

```

```

DataRec.sta(4, 0) = 1

```

```

Seek #3, recptr
Put #3, , DataRec
Seek #3, recptr
Get #3, , DataRec

```

```

If Time1Stamp <> "" Or Time2Stamp <> "" Then
    Time11Stamp = Format(Time1Stamp, "General Number")

```

```

    Time2Stamp = Format(Time2Stamp, "General Number")
    TimeForProdMin2 = Format(TimeForProdMin, "General Number")
End If
DataRec.sta(4, 1) = CostoTotalDEK
DataRec.sta(4, 2) = CostoTotalDEKReal
Debug.Print recptr, DataRec.sta(4, 0), DataRec.sta(4, 1), DataRec.sta(4, 2)
'volver a escribir el record quitando la señal para liberarlo y poniendo los datos
DataRec.sta(0, 4) = 0
Seek #3, recptr
Put #3, , DataRec
'guardar en el record de control (300) el pointer del ultimo record escrito
Seek #3, 300
Get #3, , DataRec
'***$$$***
DataRec.sta(0, 4) = recptr
'***
Seek #3, 300
Put #3, , DataRec
'preparamos para la próxima posición en el random file
recptr = recptr + 1
'-----
CheckControl:

'Chctrl = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\febrero 2011\Control final.exe"
'Abrir el file Control
'Open Chctrl For Random As #8
'Close #8
Seek #3, 1
Get #3, , DataRec
'si es 3, Control nos indica que acabemos el programa
'***$$$***
If DataRec.sta(0, 4) = 3 Then
    Close
End
End If
'si no es 0 ni 3, será 1, indicación de que Control quiere
'que entremos en pausa. Poner 2 para indicar a Control que
'hemos entrado en pausa.
'***$$$***
If DataRec.sta(0, 4) = 1 Then
    DataRec.sta(0, 4) = 2
    Seek #3, 1
    Put #3, , DataRec
End If
'***$$$***
If DataRec.sta(0, 4) >= 1 Then End
'Next one
'Se usa 'GoSub Repetir cuando uso el programa solo de esta estación
'GoTo repetir
Final:
    Close #3
    xx = xx + 1
    Module6.Estaciones
End

```

Apéndice 11 Módulo de VBA® para llenar el archivo "stacomb.dat" de la quinta estación

```

.
Type t_DataRec
sta(5, 4) As Single
End Type

'=====
ESTACION 5 nueva version
'=====

Sub Form_Load()
xx = 5
'=====

Dim DataRec As t_DataRec
Dim vueltas As Integer
Dim filename1 As String      'nombre del archivo a procesar (stacomb)
Dim Time1Stamp As Variant
Dim Time2Stamp As Variant
Dim Time11Stamp As Variant
Dim Time22Stamp As Variant
Dim TimeForProdMin As Variant
DoEvents
'-----

recptr = 1
Sheets("Sheet3").Select
    Cells.Select
    Selection.ClearContents
filename1 = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\stacombnew.dat"
Open filename1 For Random As #3 Len = Len(DataRec)
'el random file que acabamos de abrir, debe haberse creado con otro programa
'que crea un archivo de 300 records. Las primeras cinco posiciones (20 bytes)
'del record 1 y del record 300 son datos de control. Después siguen 5 grupos
'de 5 posiciones cada una, un grupo para cada estación. La primera posición
'de cada grupo se utilizará para record locking, las otras cuatro para
'escribir los datos de salida: tiempo, componentes, errores y comp. extras
'Los datos de control en el record 1 son para indicar a los programas
'que procesen, pausen o dejen de ejecutar. Los datos de control del
'record 300 son para ir registrando el último record que han escrito cada
'programa de estaciones.

```

```

Sheet3.Cells(1, 1) = "Numero"
Sheet3.Cells(1, 2) = "Sensor In"
Sheet3.Cells(1, 3) = "Sensor Out"
Sheet3.Cells(1, 4) = "Tiempo S-In"
Sheet3.Cells(1, 5) = "Tiempo S-Out"
Sheet3.Cells(1, 6) = "Tiempo estacion"
Sheet3.Cells(1, 7) = "Costo total DEK"
Sheet3.Cells(1, 8) = "Costo total DEKReal"

```

'si el sensor de entrada de la DEK cambia de estado, entonces se guarda la hora en la que cambio de estado

'en la celda sta(1,0)

'Anota el tiempo de entrada del board a la estación.

empieza:

aqui = 2

repetir:

'Anota la hora en que el sensor C detecta que hay una pieza pasando

Do Until Sheet3.Cells(aqui, 2) = 1

 mychannel = DDEInitiate("datahub", "default")

 Application.Worksheets("Sheet3").Activate

 InDEK = DDERequest(mychannel, "Channel1.PLC.Salida5")

 Sheet3.Cells(aqui, 2) = InDEK

 DDETerminate mychannel

 vueltas = vueltas + 1

 If vueltas = 150 Then

 GoTo Final

 End If

Loop

 Time1Stamp = Format(Time, "hh:mm:ss")

 Sheet3.Cells(aqui, 4) = Time1Stamp

 InDEK = 0

'Anota la hora en que el sensor D detecta que hay una pieza pasando

Do Until Sheet3.Cells(aqui, 3) = 1

 mychannel = DDEInitiate("datahub", "default")

 Application.Worksheets("Sheet3").Activate

 OutDEK = DDERequest(mychannel, "Channel1.PLC.Salida6")

 Sheet3.Cells(aqui, 3) = OutDEK

 DDETerminate mychannel

```

vueltas = vueltas + 1
If vueltas = 150 Then
    GoTo Final
End If
Loop
    Time2Stamp = Format(Time, "hh:mm:ss")
    Sheet3.Cells(aqui, 5) = Time2Stamp
    OutDEK = 0
    If Sheet3.Cells(aqui, 4) <> Sheet3.Cells(aqui, 5) Then
        TimeForProdMin = ((Sheet3.Cells(aqui, 5)) - Int(Sheet3.Cells(aqui, 5))) * 24 - ((Sheet3.Cells(aqui, 4)) - Int(Sheet3.Cells(aqui, 4))) * 24
        Sheet3.Cells(aqui, 6) = TimeForProdMin
    End If
    '=====
    'Calculo de costos del HORNO TEORICO(estacion 5)
        'Alistamiento ($/panel)        'Nitrogeno ($/panel)
CostoHORNO = ((HrSetupHORNO / PanelesDiario) * MdeOD) + (TanqueN / BoardsN) + ((AreaEdificio * Edificio) / (PanelesDiario * NumEstaciones))

        'Consumo de energía de las bombillas ($)        'Consumo de energía del A/C ($)
'Consumo de energía del A/C ($)        'Consumo de energía del compresor ($)        "Consumo de energía de las computadoras ($)
'Costo energía($ / panel)
CostoEnergiaHORNO = (((((WBombillas * NumBombillas * 270) / 1000) * KwEnergia) + (((WAC * 270) / 1000) * KwEnergia) + (((WHORNO * 270) / 1000) * KwEnergia) + (((WCompresor * 270) / 1000) * KwEnergia) + (((WComputadoras * NumComputadoras * 270) / 1000) * KwEnergia)) / (PanelesDiario * NumEstaciones))

CostoTotalHORNO = CostoHORNO + CostoEnergiaHORNO
    Sheet3.Cells(aqui, 7) = CostoTotalHORNO 'Anotar las ecuaciones de costo teorico

    'Calculo de costos del HORNO REAL (estacion 5)

        'Alistamiento ($/panel)        'Nitrogeno ($/panel)
CostoHORNOReal = ((HrSetupHORNO / PanelesDiario) * MdeOD) + (TanqueN / BoardsN) + ((AreaEdificio * Edificio) / (PanelesDiario * NumEstaciones))

        'Consumo de energía de las bombillas ($)        'Consumo de energía del A/C ($)
'Consumo de energía del A/C ($)        'Consumo de energía del compresor ($)        "Consumo de energía de las computadoras ($)
'Costo energía($ / panel)
CostoEnergiaHORNOReal = (((((WBombillas * NumBombillas * TimeForProdMin) / 1000) * KwEnergia) + (((WAC * TimeForProdMin) / 1000) * KwEnergia) + (((WHORNO * TimeForProdMin) / 1000) * KwEnergia) +

```

```
(((WCompresor * TimeForProdMin) / 1000) * KwEnergia) + (((WComputadoras * NumComputadoras *
TimeForProdMin) / 1000) * KwEnergia)) / (PanelesDiario * NumEstaciones))
```

```
CostoTotalHORNORReal = CostoHORNORReal + CostoEnergiaHORNORReal
```

```
Sheet3.Cells(aqui, 8) = CostoTotalHORNORReal
```

```
' anotar la ecuacion de costo real
```

```
'=====
```

```
aqui = aqui + 1
```

```
GoTo repetir
```

```
'-----
```

```
DataWrite:
```

```
'Aqui se implementa record locking que dada la naturaleza del proceso
```

```
'(pasa un tiempo entre que una estación escriba y escriba la siguiente)
```

```
'me parece innecesario. Pero se deja como un ejercicio de programación
```

```
'y por si alguna vez se modifica los programas para hacer cosas con
```

```
'el archivo donde haya riesgo que un programa escriba datos a la vez que
```

```
'otro actualiza.
```

```
'primero asegurar que no hay otro programa escribiendo
```

```
'en el record, o que estamos en pausa.
```

```
Seek #3, recptr
```

```
Get #3, , DataRec
```

```
tlock = 0
```

```
' For il = 1 To 5
```

```
' tlock = tlock + DataRec.sta(il, 0)
```

```
' Next il
```

```
'si la suma (tlock) no es cero, hay alguien usando el record
```

```
'volver a leerlo hasta que quede libre
```

```
If tlock > 0 Then GoTo DataWrite
```

```
'poner la señal de que queremos usar el record
```

```
DataRec.sta(4, 0) = 1
```

```
Seek #3, recptr
```

```
Put #3, , DataRec
```

```
Seek #3, recptr
```

```
Get #3, , DataRec
```

```
DataRec.sta(5, 1) = CostoTotalDEK
```

```
DataRec.sta(5, 2) = CostoTotalDEKReal
```

```

Debug.Print recptr, DataRec.sta(5, 0), DataRec.sta(5, 1), DataRec.sta(5, 2)
'volver a escribir el record quitando la señal para liberarlo y poniendo los datos
DataRec.sta(0, 4) = 0
Seek #3, recptr
Put #3, , DataRec
'guardar en el record de control (300) el pointer del ultimo record escrito
Seek #3, 300
Get #3, , DataRec
'***$$***
DataRec.sta(0, 4) = recptr
'***
Seek #3, 300
Put #3, , DataRec
'prepararnos para la próxima posición en el random file
recptr = recptr + 1
'-----
CheckControl:
'Chctrl = "C:\Users\Enrique\Desktop\paper\eventdata may4\Mach\eventdata\ip3\febrero 2011\Control final.exe"
'Abir el file Control
'Open Chctrl For Random As #8
'Close #8
Seek #3, 1
Get #3, , DataRec
'si es 3, Control nos indica que acabemos el programa
'***$$***
If DataRec.sta(0, 4) = 3 Then
Close
End
End If
'si no es 0 ni 3, será 1, indicación de que Control quiere
'que entremos en pausa. Poner 2 para indicar a Control que
'hemos entrado en pausa.
'***$$***
If DataRec.sta(0, 4) = 1 Then
DataRec.sta(0, 4) = 2
Seek #3, 1
Put #3, , DataRec

```


End If

\$\$\$

If DataRec.sta(0, 4) >= 1 Then End

Final:

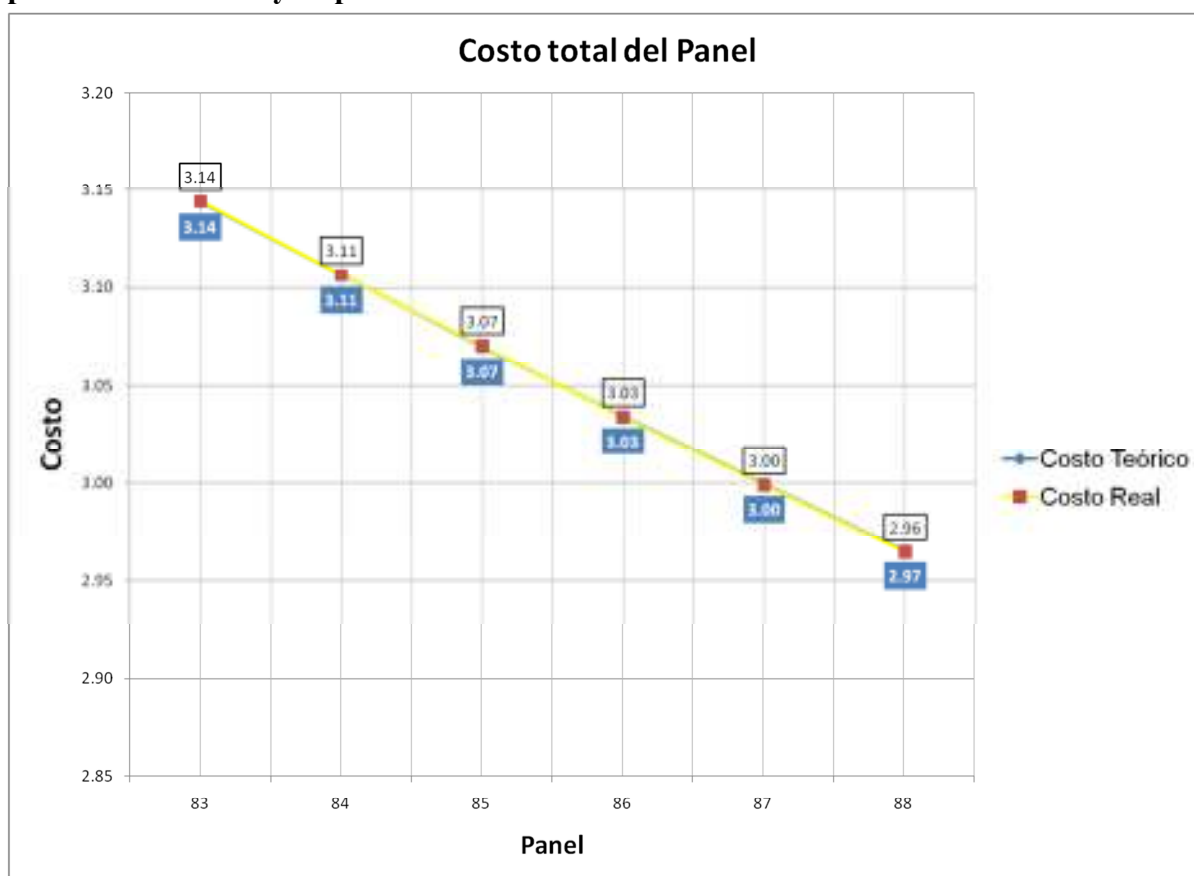
Close #3

xx = xx + 1

Module6.Estaciones

End Sub

Apéndice 12 Gráfica y Reporte final



Gráfica 1 Gráfica final

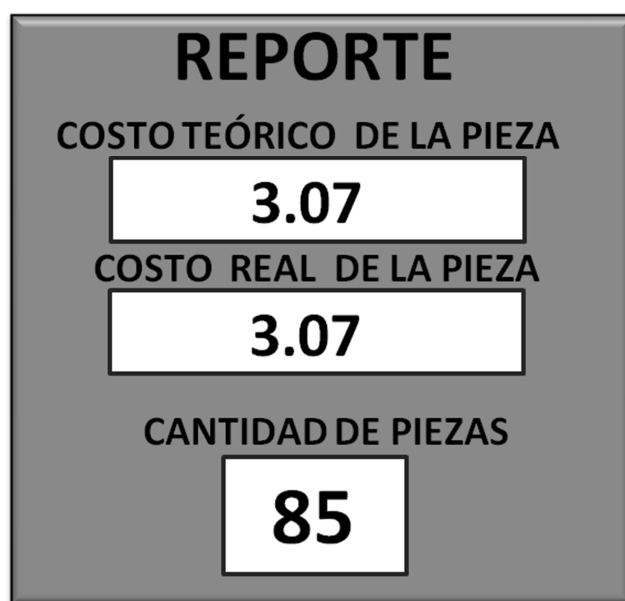


Figura 17 Reporte final