

# Un modelo para desarrollar sistemas tipo SCADA en entornos productivos

José Luis Ochoa-Hernández, Mario Barcelo-Valenzuela, Federico Miguel Cirett-Galán,  
Ramón Alberto Luque-Morales

Universidad de Sonora, Department of Industrial Engineering, Hermosillo, Sonora,  
Mexico

{joseluis.ochoa, mbarcelo, fcirett, rluque}@industrial.uson.mx

**Resumen.** En las últimas décadas se ha hecho indispensable el desarrollo de sistemas industriales que deben procesar gran cantidad de datos en breves periodos de tiempo. Estos sistemas se desarrollan poco en México, por un lado son complejos y especializados y por otro, deben ser claros y prácticos para que puedan ser utilizados en diversas áreas como es la de producción, ya que estos deben controlar la maquinaria en tiempo real y reflejar el estado de la misma en pantallas ubicadas a distancia, también, deben ser capaces de obtener información histórica para la toma de decisiones concretas en forma de reportes específicos según se requiera. En este artículo se presenta un modelo creado para desarrollar sistemas tipo SCADA el cual fue empleado en dos proyectos 1) trituramiento y pulverización y 2) pulimento y limpieza, ambos de forma automatizada y de materia prima hecha a base de vidrio reciclado. El sistema se validó y los resultados obtenidos fueron satisfactorios, se espera que sirva de guía para desarrollos similares.

**Palabras clave.** Sistemas industriales de control, sistemas SCADA, desarrollo de sistemas de información, sistemas de soporte para la toma de decisiones.

## A Model to Develop SCADA Systems in Productive Environments

**Abstract.** In the last decades, the development of industrial systems capable of processing large amounts of data in short time periods has become crucial. These systems are narrowly developed in Mexico due to their focused nature and complexity. They must be practical to use in diverse areas, such as production, where the system remotely controls and reports machine metrics in real time and reflect the status of it, on screens located on distance; they should also be capable of providing

historical data when required. This article presents a model created to develop SCADA systems which was used in two projects 1) crushing and pulverization and 2) polishing and cleaning, both automated and raw material made from recycled glass. The system was validated and the results obtained were satisfactory, is expected to serve as a guideline for similar projects.

**Keywords.** Industrial control systems, SCADA systems, development of information systems, support systems for decision making.

## 1. Introducción

En las últimas décadas las empresas están reconociendo que trabajar con la mayor cantidad de información tanto histórica como actual, les ayuda a obtener mejores resultados, ya sea para realizar mejores planeaciones o para tomar mejores decisiones [5]. Existen diversos factores que están haciendo que las empresas sean mucho más competitivas, uno de ellos es la globalización, lo que conlleva al uso de la tecnología informática como recurso organizacional [6] y [11]. De hecho, aquellas empresas que tengan como Visión la de crecer, son aquellas que emplearán a la tecnología como una ventaja competitiva y, dependiendo del nivel de tecnología que apliquen solo se mantendrán o bien realmente podrán trabajar de manera global, convirtiéndose esta en un recurso de base operacional de la empresa [4].

Los gerentes o responsables de la toma de decisiones, han comprendido que la información no es solo un elemento de conducción empresarial o un elemento histórico que se genera y se guarda,

sino que es un elemento de gran valor que alimenta a los negocios y que puede ser uno de los tantos elementos críticos que pueden determinar el éxito o fracaso de los negocios [15, 2, 5].

Normalmente las empresas, si guardan cierta información, como la de los clientes, los distribuidores, proveedores, financiadores, empleados, etc. [9]. Sin embargo, por si sola, esta no genera valor, lo hace hasta que se analiza, se procesa y cataloga, es decir, hasta que se rentabiliza.

Para algunas empresas, el análisis de la información, se está volviendo un elemento clave, para otras, les está ayudando a sobrevivir más tiempo en el mercado. La información que obtienen les ayuda a responder a sus entornos con rapidez y les permite evolucionar con el paso del tiempo, hoy en día aquellas empresas que no son eficientes, ahorradoras y no emplean tecnología informática están desapareciendo.

Sin embargo, el simple hecho de usar tecnología no significa “éxito”, empresas que han apostado por el uso de la tecnología informática, han desaparecido o simplemente no han obtenido el beneficio esperado y esto se debe a la mala aplicación de esta [7] y/o a la mala calidad del software [26] ya que un software que produce un error insignificante puede causar grandes problemas, un redondeo de cifras mal aplicado, duplicidad de un registro, fecha mal registrada, etc.

En el sector industrial se maneja una gran cantidad de información que es muy importante, como son entre otros, los datos generales de control, de automatización y mantenimiento de la producción, los datos operativos de la empresa, etc. así como, los datos de las maquinarias de forma individual y/o por líneas de producción, las cuales hacen que se mantenga y se produzca lo necesario en óptimas condiciones.

Los Sistemas de Control de Supervisión y Adquisición de Datos (SCADA) son un tipo de sistemas de información que son introducidos para mejorar los procesos de negocio [14], generan una gran cantidad de datos especializados provenientes del área de producción [20], los cuales son mostrados a diversos usuarios, como operadores, supervisores de control de calidad, mantenimiento, etc. Estos sistemas se encuentran catalogados dentro de los Sistema de Control

Industrial (SCI) los cuales incluyen adicional al SCADA, Sistemas de Control Distribuido (DCS), los cuales controlan diversas unidades de proceso y algunos otros como los Controladores Lógicos Programables (PLC) que se centran en llevar y mantener el control y automatización y que a menudo se encuentran en los sectores industriales e infraestructuras críticas, ya que su precisión y eficacia es muy elevada [29].

El modelo propuesto en esta investigación se implementó en dos proyectos en una empresa del noroeste de México dedicada al reciclado del vidrio.

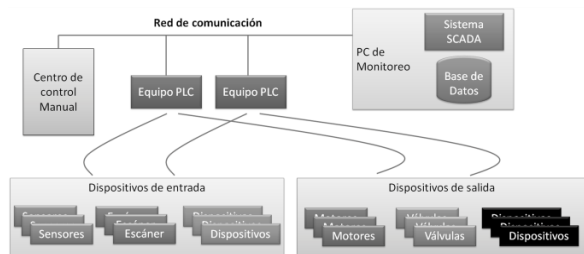
Este artículo se estructura de la siguiente forma: en la segunda parte se presenta el marco teórico acerca de los sistemas SCADA y la metodología de desarrollo, en la tercera se describe de forma practica el modelo propuesto, se detallan sus 4 elementos básicos que son: la investigación preliminar, la determinación de los requerimientos, el proceso de desarrollo y la implementación y evaluación; en la cuarta se presentan los resultados de la investigación divididos en categorías y finalmente en la quinta, se exponen las conclusiones.

## 2. Marco teórico

El desarrollo de sistemas no es una tarea fácil o simple de hacer, razón por la cual surgieron diversas metodologías para desarrollar software como lo explica [29], donde cada una garantiza el éxito al seguir sus propias directrices, sin embargo, estas son diferentes, ya que existen metodologías para el desarrollo de proyectos grandes y duraderos, así como metodologías ágiles para el desarrollo de proyectos extensos, con limitación de tiempo y cambiantes [16].

Los sistemas que trabajan con automatización y control agregan una variable más a la dificultad del desarrollo de sistemas, que es la “dependencia”, ya que interactúa con diversos instrumentos (sensores o medidores) recolectores de datos, los cuales sirven de entrada al sistema y otros dispositivos (actuadores) que ejecutan acciones indispensables basadas en el procesamiento de los datos de producción.

A este tipo de sistemas se les conoce como sistemas tipo SCADA, que son aplicaciones



**Fig. 1.** Representación esquemática del funcionamiento de un Sistema SCADA

informáticas especialmente diseñadas para funcionar sobre computadoras de control de la producción, las cuales tienen acceso a esta área mediante una comunicación digital que se basa en diversos instrumentos y/o actuadores. Así mismo, incluyen una interfaz gráfica de alto nivel operacional, donde el operador a través de esta puede supervisar parte o la totalidad de la producción y al mismo tiempo interactuar con parte o la totalidad de esta en caso de que se requiera.

Los sistemas SCADA son comúnmente usados en industrias eléctricas (CFE), en las plantas tratadoras de aguas residuales (CONAGUA), de extracción de petróleo y / o gas natural (PEMEX), en empresas químicas, de transporte, farmacéuticas, de producción de papel, alimentos y bebidas como es el caso de [27] así como en plantas nucleares [17].

Diversos autores como [17] y [19] comentan que estos sistemas han evolucionado de una forma notable, desde implementaciones con componentes discretos y cableados hacia dispositivos inteligentes e inalámbricos distribuidos geográficamente, aumentando su capacidad de configuración y los medios de comunicación mediante módulos y protocolos estándar. Lo cual provocó pasar de implementaciones en general propietarias y cerradas (por ciertas marcas), a implementaciones de arquitectura abierta, basada en tecnología digital, con estándares de tecnología de información de dominio público, lo que ha hecho que cualquier persona de cualquier lugar interesada, pueda manipular estos dispositivos bajo desarrollos propios.

Existe un concepto que explica esta evolución, se conoce como modelado de variabilidad [10], el

cual se hace mediante diversas técnicas, una de ellas es la COVAMOF la cual permite modelar los puntos de variación, variantes y dependencias de los diversos modelos de abstracción de una Línea de productos de software (LPS) [8], sin embargo, en [30] se define la variabilidad como la “habilidad de cambiar o personalizar un sistema”, que es lo que se plantea en este artículo.

El proceso funcional de los sistemas SCADA abarca desde la recolección de los datos hasta la activación de una acción en planta, llevando los datos a la unidad de control, para que estos sean analizados e interpretados para luego ser mostrados en una o varias pantallas al operador y de esta manera según se requiera, se permita la interacción humano-máquina tanto en planta como en campo [1].

Para ello se requieren los Controladores Lógicos Programables (PLC) que son los elementos que sirven de intermediarios entre los sistemas SCADA y los dispositivos conectados a la maquinaria los cuales se encargan de recibir datos y enviar señales para lograr el control y automatización de los equipos. [21] los define como: “un dispositivo operado digitalmente, que usa una memoria para el almacenamiento interno de instrucciones con el fin de implementar funciones específicas, tales como lógica, secuenciación, registro y control de tiempos, conteo y operaciones aritméticas, para controlar a través de entradas/salidas digitales o analógicas, varios tipos de máquinas o procesos.”

Para [24] los sistemas SCADA presentan un conjunto de características que agrupan todo el proceso de control y automatización, donde se pueden resaltar además del proceso base de adquisición y almacenado de datos, representación gráfica y animada de variables de proceso y su monitorización por medio de alarmas, supervisión de las variables de control, ejecución de acciones de control para modificar la evolución del proceso, etc., una arquitectura abierta y flexible con capacidad de ampliación y adaptación, conectividad con otras aplicaciones y bases de datos, locales o distribuidas en redes de comunicación, explotación de los datos adquiridos para gestión de la calidad, control estadístico, gestión de la producción y gestión administrativa y financiera.

Un ejemplo de arquitectura de los sistemas SCADA funcionando en conjunto con un PLC se puede ver en la figura 1, donde se muestran los elementos básicos generales para la supervisión y control de cualquier maquinaria o planta que se conecte a las terminales del PLC. Se tiene un centro de control donde se manejan algunas de las máquinas o procesos esenciales del ciclo de producción, incluidos el botón de emergencia, que al ser presionado realiza un paro de emergencia de toda la maquinaria y un control de encendido que controla el ciclo de producción, los equipos PLC claves y esenciales en el proceso de automatización, los sensores que interactúan con el controlador PLC y que están conectados a las maquinarias que se deseen controlar, la PC de monitoreo que contendrá el sistema SCADA, así como una base de datos que interactúe con el sistema informático de soporte a la toma de decisiones.

En este caso, todos los elementos juegan un papel importante en el proceso de control y automatización, sin embargo, los dispositivos de entrada, el controlador y los dispositivos de salida se consideran los elementos principales del proceso.

### 3. Modelo propuesto

El proceso de creación de un sistema SCADA propuesto en este artículo se puede observar en la figura 2, la elaboración de este modelo es el resultado de la investigación, conocimiento y experiencia del equipo de trabajo, se elaboró sistemáticamente para que pueda ser seguido por cualquier otro equipo de desarrollo y pueda ser adaptado a sus necesidades.

El desarrollo del sistema informático fue creado en base al ciclo de vida clásico de desarrollo de software el cual está definido en [25], este ciclo es el que utilizan los analistas, diseñadores y programadores para desarrollar e implantar productos de software.

El modelo aquí propuesto cuenta con las siguientes etapas 1) Investigación preliminar, 2) Determinación de los requerimientos del sistema, 3) Proceso de desarrollo (el cual incluye el Diseño, Desarrollo y Pruebas al sistema, las cuales asegurarán la calidad del Software [26]) y 4)

Implementación y evaluación. A continuación, se explica la metodología empleando el caso práctico “Producción de materias primas a base de vidrio reciclado”:

#### 3.1. Investigación preliminar

Para cumplir con la etapa 1 del modelo, se llevó a cabo una investigación de las posibles tecnologías a utilizar en el desarrollo de este proyecto en base al orden mostrado:

1. Documentación del proyecto en general para conocer las necesidades reales de este: Se deben realizar visitas empresariales para conocer el funcionamiento del proceso y conocer que es lo que se quiere desarrollar, se debe estudiar la documentación proporcionada del proyecto, así como realizar entrevistas a los diversos actores.
2. Características principales del sistema operativo donde se ejecutará el sistema: elaborar un estudio que contemple las Fichas Técnicas, la Disponibilidad, el Soporte Técnico, los Costos (licencias), el Mantenimiento que requiere, así como la seguridad y la operatividad.
3. Características de los Sistemas y Administradores de bases de datos: analizar los tipos de datos que se almacenaran, así como su cantidad, de igual forma el desempeño esperado al realizar consultas a la base de datos.
4. Características de las metodologías de desarrollo: estudiar las propuestas de trabajo de cada una, tomando en cuenta el tiempo de desarrollo, los posibles cambios y su repercusión en el sistema.
5. Características de los modelos de desarrollo: analizar las propuestas de trabajo, así como las ventajas y desventajas que presenta cada modelo.
6. Características de los Lenguajes de Programación: tomar en cuenta la popularidad de cada lenguaje, sus paradigmas, la facilidad de uso, la frecuencia de actualización y mantenimiento, las mayores ventajas y desventajas que estos pueden proporcionar.
7. Características de los controladores existentes en el mercado: analizar los

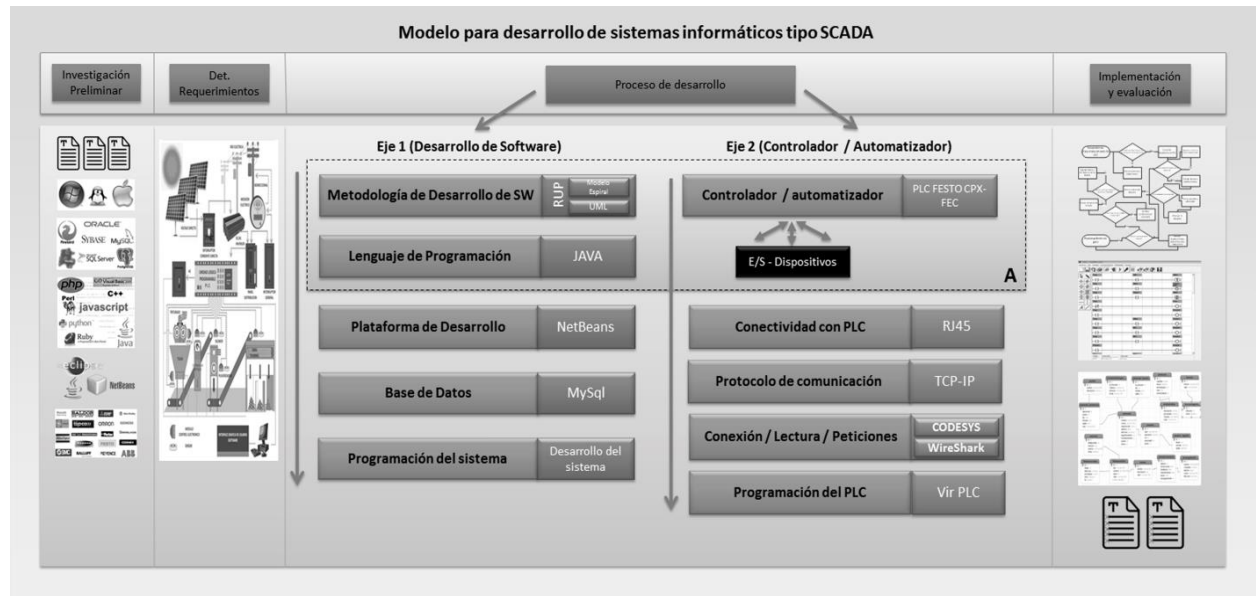


Fig. 2. Modelo para desarrollar sistemas informáticos tipo SCADA

diferentes tipos de controladores que existen (PLC, Relés, Microcontroladores) y de cada uno de ellos, algunos conceptos básicos como ficha técnica, la disponibilidad en la región, el soporte técnico en la región, costos de cada controlador y de sus equipos básicos o suplementarios disponibles, servicios adicionales (software de gestión, manuales) y sobre todo la seguridad.

### 3.2. Determinación de los requisitos y requerimientos

El primero elemento que se define son los requisitos, ya que sin estos no es posible medir el alcance del proyecto y las fronteras de este. El proceso para identificar los requisitos se basa en 3 pasos: 1) comprender el problema, 2) comprender el dominio y 3) comprender el modelo de negocio.

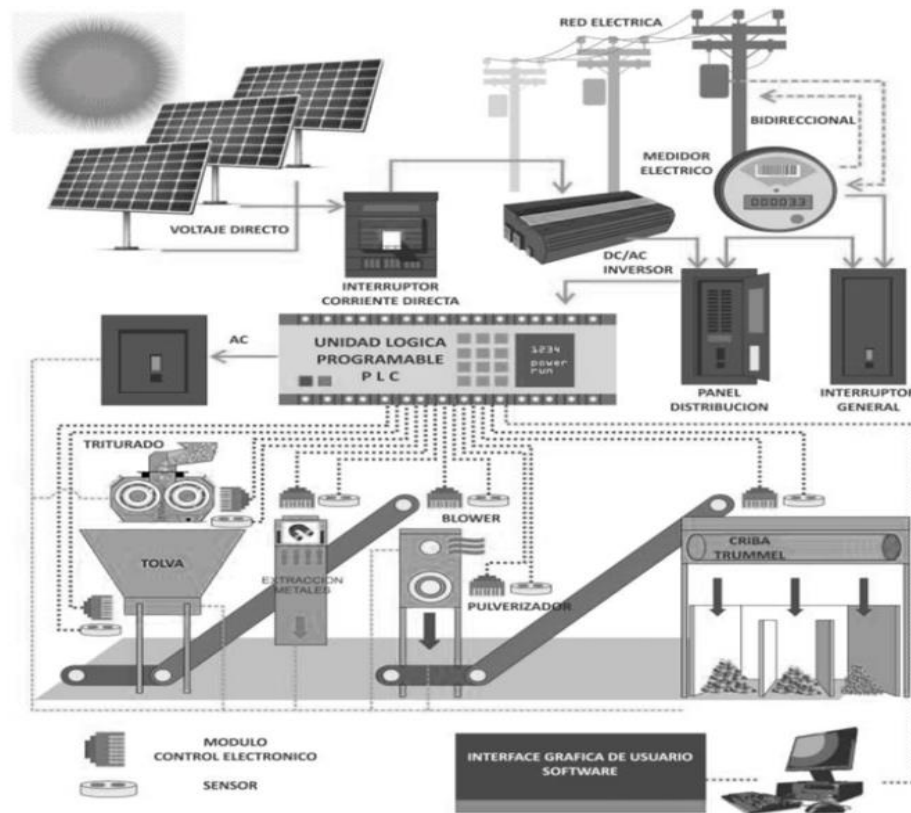
Algunos requisitos pueden estar definidos desde el principio como restricciones y algunos otros se obtienen gracias a los 3 pasos anteriores, para ello se tomaron en cuenta las necesidades y objetivos que contemplaba el cliente, así como las condiciones en las que operaría la maquinaria y el equipo propuesto de control.

Para este caso práctico el objetivo general es controlar la producción de arena y grava de vidrio

de diversos tamaños y colores y en base a la infografía definida (ver en la figura 3) se determinó el proceso de producción (elemento clave para identificar los requerimientos).

El proceso de producción se explica en los siguientes pasos: 1) Se recolecta el vidrio ya sea mediante compra directa o en base a recolectores, 2) se clasifica de forma manual por colores, el vidrio que no se clasifica sirve para producir vidrio multicolor, 3) se elimina manualmente la basura muy grande, 4) comienza el proceso de triturado, el vidrio es introducido en la máquina de triturado, la salida se recolecta en una tolva, la cual dosificará la salida a la banda transportadora, 5) durante su avance en la banda, pasa por un extractor de metales, que retira fichas y otros objetos metálicos, 6) el vidrio continua el proceso pasando por un *blower*, el cual elimina la basura más pequeña, como hojas, etiquetas, etc., 7) el vidrio pasa por una pulverizadora, cuyo objetivo es reducir a polvo parte del vidrio, 8) los trozos de vidrio y polvo llegan a una criba donde esta se clasificará por tamaños, obteniendo el producto final.

Todo este proceso deberá ser alimentado mediante energía solar y controlado mediante un PLC y un sistema SCADA que muestre el monitoreo y control de la producción.



**Fig. 3.** Infografía de la producción de grava y polvo de vidrio mediante energía solar, controlado por un sistema informático

Algunos de los requisitos más importantes se comentan a continuación:

- Obtener la información de los sensores conectados al PLC para llevar un control de las fallas, de las anomalías, de la producción, así como el tiempo muerto, tiempo de producción, tiempo de manteniendo, tiempo de producción, material para reprocesar, el número de veces que sucedió algún evento, hora de encendido y apagado de las máquinas, etc.
- Existirán botones de paro de emergencia en cada una de las máquinas, así como en el sistema informático.
- El sistema será intuitivo y fácil de usar.
- El sistema registrará usuarios para llevar un control de personal.
- El sistema contendrá alarmas visuales y sonoras tanto en planta como en la computadora de control.
- El sistema generará reportes de producción, indicadores de desempeño y errores, empleando gráficas para su fácil entendimiento.

### 3.3. Proceso de desarrollo

Una vez que se identifican los requisitos y se definen los requerimientos, etapa laboriosa y de suma importancia [23], el proceso de desarrollo de software es el que continua con el modelo propuesto. El cual consta de 2 ejes principales: el Eje 1 (Desarrollo de software) y el Eje 2 (Controlador / Automatizador).

Se requiere prestar una atención especial al bloque 'A' ya que contiene los elementos

principales por definir antes de proceder con el resto de elementos, estos son: la “*Metodología de Desarrollo*”, el “*lenguaje de programación*” y el “*controlador / automatizador*” incluidos los dispositivos de Entrada / Salida.

El primer eje “**Eje 1 (Desarrollo de software)**”. menciona los elementos necesarios para desarrollar el software, el cual incluye la elección de una metodología de desarrollo de software, la elección de un lenguaje de programación, la elección de una plataforma de desarrollo de software, la elección de una base de datos y la programación del sistema

El segundo Eje “**Eje 2 (Controlador / Automatizador)**”. menciona los elementos necesarios para trabajar con el equipo de control y automatización del proyecto a controlar, este incluye la elección del controlador o automatizador con los dispositivos de E/S, la elección de la conectividad con el controlador elegido, la elección de los protocolos de comunicación, la elección de la librería de “conexión – lectura – peticiones” del lenguaje elegido con el controlador elegido, – elemento clave para lograr recuperar y enviar los datos, al controlador– y la propia programación del PLC que moverá a la maquinaria.

Estas actividades (ver figura 2), se realizan de forma paralela conforme se avanza con el proceso de desarrollo, sin embargo, no están limitadas a realizarse en pareja, por ejemplo, al momento de elegir la *Plataforma de desarrollo*, no es necesario elegir la *conectividad con el PLC*, son actividades que se tienen que realizar para poder implementar, pero no es necesario elegir las al mismo tiempo.

Cada uno de los elementos que se muestran en este modelo son *variables* y *dependientes* de la experiencia, la tecnología y la capacidad del equipo de desarrollo de software.

Cabe aclarar que algunos equipos de control y/o de automatización normalmente son caros, y esto se debe a que el uso común de la automatización es empleado para realizarán ciertas tareas normalmente repetitivas y en ocasiones de interacción de forma simultánea con

otras máquinas o componentes que hacen que las pruebas en entornos reales no sean factibles y/o seguras de realizarse, por lo que la simulación se vuelve un componente fundamental y ampliamente recomendado. De hecho [32] comentan que en el desarrollo de software de control, el costo total de desarrollo del sistema, ha llegado a alcanzar una proporción de más del 50% de los costos que involucran el desarrollo de los sistemas industriales.

Debido a que la programación del PLC es una de las actividades que tiene gran importancia en la automatización, existe software de simulación virtual que reduce significativamente los costos, un ejemplo de este software es la plataforma ProWORX 32<sup>1</sup> que abarca diversas marcas de controladores, el software Lav 1.9<sup>2</sup>, específico para controladores Siemens y VirPLC<sup>3</sup> también multimarca (el cual fue elegido en este proyecto por incluir el controlador Festo modelo CPX-FEC), incluye las lógicas de control (con lo que se hará la programación) para ayudar a resolver y validar la problemática presente, de igual forma [18, 33] comentan que esta simulación ayuda a desarrollar prototipos de forma rápida.

Existen 2 actividades que requieren mayor atención, la elección de metodología de desarrollo de SW y la obtención de datos desde el PLC, por tal razón se describen estas actividades explicando porque se usó la metodología RUP para el desarrollo de SW y como se llevó a cabo la Comunicación en 2 vías con él PLC.

### 3.3.1. Explicación del proceso de desarrollo de SW siguiendo la metodología RUP

Las metodologías de desarrollo de sistemas, están para cumplir de la mejor forma los objetivos sin dejar nada olvidado dando un seguimiento adecuado. En este desarrollo se siguió RUP, cuyo objetivo de esta metodología según [22] es establecer planes razonables para la ejecución de ingeniería de software y para la administración de proyectos de software.

<sup>1</sup> ProWorx, [http://www.schneider-electric.co.cr/costa-rica/es/productos-servicios/automatizacion-control/oferta-de-productos/presentacion-de-rango.page?p\\_function\\_id=18&p\\_family\\_id=236&p\\_range\\_id=546](http://www.schneider-electric.co.cr/costa-rica/es/productos-servicios/automatizacion-control/oferta-de-productos/presentacion-de-rango.page?p_function_id=18&p_family_id=236&p_range_id=546) Consultado en línea, Septiembre de 2016.

<sup>2</sup> Laboratorio de Automatización Virtual, <http://isa.uniovi.es/genia/spanish/download/index.htm> Consultado en línea, Septiembre de 2016,

<sup>3</sup> Virtual PLC simulador, <http://www.xtec.cat/~aferrer8/virplc.htm> Consultado en línea, Septiembre de 2016.



RUP es un modelo que identifica cuatro fases discretas en el proceso de software como menciona [28], a continuación, se describen brevemente estas fases aplicadas al proyecto:

1. **Concepción:** en esta fase se requiere definir un objetivo empresarial, el cual se definió como “desarrollo de un sistema SCADA que controle la producción de materia prima a base de vidrio reciclado en tiempo real, con la funcionalidad de generar de informes prácticos para la toma de decisiones”, para ello se definen los actores involucrados que interactuarán con el sistema y se valora la aportación del sistema hacia la empresa.
2. **Elaboración:** en esta fase define el lenguaje a utilizar basado principalmente en sus características y en la experiencia del equipo, para este caso será JAVA (según Tiobe<sup>4</sup> y Universia Noticias<sup>5</sup>, es el lenguaje más usado y con mayor cantidad de documentación disponible, así como es reconocido por su legibilidad y simplicidad), se requiere definir también la plataforma de desarrollo, lo cual se hace en base al lenguaje elegido y a que presente mayores ventajas para el desarrollo, en este caso será NetBeans. Se define también el manejador de base de datos, el cual se basa en la capacidad de procesamiento, capacidad de almacenamiento, cantidad de usuarios y cantidad de recursos demandados estimados, para el caso práctico será MySQL. De forma paralela a la que se definen los elementos de SW, se requiere definir el controlador de la maquinaria a utilizar, el cual será FESTO CPX-FEC basándose principalmente en la cantidad de entradas y salidas necesarias para el proyecto a desarrollar, (que en este caso se contemplan 3 etapas de manera general), se elige el tipo de conectividad en base a las conexiones que el controlador provee y las características de comunicación, en este caso será por cable de Red y el protocolo de comunicación, que depende del fabricante del controlador y el software empelado, en este

caso será (TCP-IP). Se define también la arquitectura del sistema (según las necesidades del proyecto), se diseña el plan del proyecto (por etapas (en base a la metodología) en este caso fue el modo espiral) y se identifica el riesgo principal de la aplicación (tarea básica del análisis), por último se identifica la comunicación en 2 vías con el PLC.

En caso de que se requiera un ajuste de los requisitos o requerimientos, después de analizar todos los elementos, se reinicia el proceso, se evalúa y valoran las situaciones y se prosigue con el siguiente paso (Construcción).

- 3 **Construcción:** Se desarrolla del sistema por etapas según RUP, empelando el modelado iterativo, incluye el diseño del sistema definido mediante UML ya que es el lenguaje de interacción, comunicación y documentación estándar en el mundo como lo comenta [3], esta es la fase más demandante de RUP. De forma paralela, se debe realizar la programación del PLC, así como la comunicación de 2 vías con él PLC (se incluye un apartado concreto sobre de la conexión, lectura y peticiones con el PLC, ver “2.3.2 Comunicación de 2 vías con él PLC”).
- 4 **Transición:** en esta fase se transfiere el sistema a la empresa, previo a esto se solicitó como requisito que se tuvieran listas todas las conexiones físicas de los sensores y actuadores con el PLC, para poder llevar a cabo la secuencia de pruebas definidas. Con las pruebas realizadas se obtiene retroalimentación del sistema y se lleva a cabo un plan de mantenimiento dejando el sistema funcionando en un ambiente real.

### 3.3.2. Comunicación de 2 vías con él PLC

El camino más fácil y práctico es lograr la comunicación mediante el software provisto por el fabricante del PLC, en nuestro caso FESTO, es decir, que el SW de FESTO sea el intermediario que permita recuperar los datos del controlador y enviar señales a los actuadores, sin embargo, este software no es libre para poder manipularlo

<sup>4</sup> TIOBE – The software quality company, <http://www.tiobe.com/tiobe-index/> Consultado en Agosto 2016

<sup>5</sup> Universia noticias, <http://noticias.universia.com.ar/consejos-profesionales/noticia/2016/02/22/1136443/conoce-cuales->

[lenguajes-programacion-populares.html](http://lenguajes-programacion-populares.html) Consultado en Agosto 2016



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.2.49	192.168.2.50	TCP	62	15590->15590 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
2	0.000816000	192.168.2.50	192.168.2.49	TCP	62	1200->15590 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
3	0.000881000	192.168.2.49	192.168.2.50	TCP	54	15590->1200 [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.004588000	192.168.2.49	192.168.2.50	TCP	80	15590->1200 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=26
5	0.005639000	192.168.2.50	192.168.2.49	TCP	60	1200->15590 [ACK] Seq=1 Ack=27 Win=5840 Len=0
6	0.006441000	192.168.2.50	192.168.2.49	TCP	80	1200->15590 [PSH, ACK] Seq=1 Ack=27 Win=5840 Len=26
7	0.071235000	192.168.2.49	192.168.2.50	TCP	78	15590->1200 [PSH, ACK] Seq=27 Ack=27 Win=64214 Len=24
8	0.072211000	192.168.2.50	192.168.2.49	TCP	78	1200->15590 [PSH, ACK] Seq=27 Ack=51 Win=5840 Len=24
9	0.073819000	192.168.2.49	192.168.2.50	TCP	80	15590->1200 [PSH, ACK] Seq=51 Ack=51 Win=64190 Len=26
10	0.076223000	192.168.2.49	192.168.2.50	TCP	54	15590->1200 [FIN, ACK] Seq=77 Ack=51 Win=0 Len=0
11	0.076281000	192.168.2.49	192.168.2.50	TCP	54	15590->1200 [RST, ACK] Seq=78 Ack=51 Win=0 Len=0

Frame 2: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0  
 Ethernet II, Src: Festo\_43:ea:95 (00:0e:f0:43:ea:95), Dst: Samsung\_8f:dc:0c (e8:11:32:8f:dc:0c)  
 Internet Protocol Version 4, Src: 192.168.2.50 (192.168.2.50), Dst: 192.168.2.49 (192.168.2.49)  
 Transmission Control Protocol, Src Port: 1200 (1200), Dst Port: 15590 (15590), Seq: 0, Ack: 1, Len: 0

0000 e8 11 32 8f dc 0c 00 0e f0 43 ea 95 08 00 45 00 ...2....C....E.  
 0010 00 30 00 00 00 40 00 0e b5 14 c0 a8 02 32 c0 a8 ...0..0..2..  
 0020 02 10 04 b0 3c 88 ca 75 e9 ff be a9 ff 12 70 12 ...1..<..u.....P.  
 0030 16 00 b3 24 00 00 02 04 05 b4 03 04 02 .....S.....

Fig. 4. Ejemplo de datos recibidos del PLC

mediante código externo, en estas situaciones es cuando se requiere tomar el control mediante una comunicación con el PLC de forma externa [31].

Los pasos básicos que se deben seguir para realizar la comunicación son los siguientes, definir la marca y el modelo, ya que cada PLC presenta distintas interfaces de comunicación entre el PLC y la computadora, en general la gran mayoría de PLC cuenta con salida Ethernet y/o serial. Lo segundo es definir el SW para conectarse con el PLC, puede ser alguna librería existente o una de desarrollo propio.

Para este caso se utilizó el puerto Ethernet con el protocolo de comunicación TCP/IP, y se tomó una librería como base para modificarla de acuerdo a las necesidades propias de control.

La modificación de la librería incluyó identificar ciertos elementos básicos, por ejemplo, se identificó que la obtención de datos se realiza mediante tramas de comunicación de bytes que existen entre el PLC y los sensores, para poder tener acceso a esas tramas se requirieron usar dos programas los cuales deben estar instalados en la computadora (PC). Uno de ellos es Wireshark que es el analizador de protocolos de red más importante del mundo<sup>6</sup> y CODESYS que es una plataforma de software especialmente diseñada para cumplir con los diferentes requisitos

de muchos proyectos de automatización industrial modernos<sup>7</sup>.

Se identificó que para poder obtener las tramas del PLC es necesario mantener abierta la conexión entre el PLC y la computadora, para ello se empleó el programa CODESYS. Lo primero que se obtuvo fueron distintas series de datos, las cuales se tuvieron que analizar para poder entender el protocolo en el que se comunicaba la PC y el PLC (algunas de las pruebas realizadas se muestran en la figura 4).

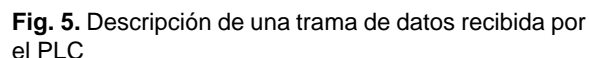
Se tuvo que realizar un proceso de conversión para lograr una mejor comprensión de la información recibida, esto fue convertirlas de bytes a hexadecimal.

Para lograr entender la estructura general de las tramas de datos, fue necesario realizar una documentación de las distintas tramas con distintos escenarios. Los cuales son:

- Obtener las tramas del PLC con distintas entradas y salidas estando los sensores encendidos y apagados.
- Realizar lecturas antes y después de iniciar la aplicación Codesys.
- Realizar lecturas antes y después de iniciar la aplicación interna del PLC.
- Realizar lecturas mientras se apaga y enciende una sola entrada del PLC.

<sup>6</sup> Wireshark, <https://www.wireshark.org/> Consultado en Septiembre 2016.

<sup>7</sup> CODESYS, <https://www.codesys.com/> Consultado en Septiembre 2016.

[illegible]

Otro proceso que fue identificado de forma adicional a mantener abierta la comunicación fue que para recibir datos, se requería enviar una trama de datos inicial, seguida de la petición al PLC para poder obtener respuestas, ya que si no se envía esta petición, no se obtenían repuestas.

Trama de datos estática PC → PLC

Trama de datos estática PLC → PC

Y para solicitar al PLC el estatus de sus componentes (es decir, los sensores) se identificó la siguiente trama:

Petición de la computadora al PLC:

```
cccc01000200000000000000000000000000001000  
000230200005005
```

```

cccc0100400000000000000000100000000000
000070000000000005000000f5717e54010101x10
1x301x501x201x401x60100012d0000000e009000
037000000013635000001000110010000fc12000
      08c0200000163b40000

```

Se verifica que el valor del socket sea mayor a un cierto número que representa el sensor, por ejemplo el valor 128 = sensor 8, el valor 64 = sensor 7, el valor 32 = sensor 6, ....., el valor 1 = sensor 1, cabe mencionar que si se cumple la condición de uno de los sensores se resta la cantidad contra la que fue comparada para obtener el estatus de los demás sensores; un ejemplo de esto se puede apreciar en las reglas definidas que son las siguientes:

1. Status Socket 1 led 8 = **Si** (Valor) >= 128  
**entonces** Socket 1 led 8=true, Valor = Valor – 128 **si no** Socket 1 led 8=false
2. Status Socket 1 led 2= **Si** (Valor) >= 2  
**entonces** Socket 1 led 2= true, Valor = Valor – 2 **si no** Socket 1 led 2= false
3. Status Socket 1 led 1= **Si** (Valor) >= 1  
**entonces** Socket 1 led 1= true, Valor = Valor – 1 **si no** Socket 1 led 1= false

Aquellos elementos que tengan un valor de "True", significa que están activos.

### 3.4. Implementación y evaluación

La implementación tiene diversos significados, uno de ellos es la instalación de una aplicación informática y otro puede ser la realización o la ejecución de un plan, idea, modelo científico, diseño, especificación, estándar, algoritmo o política<sup>8</sup>, el contexto que será empleado en este

<sup>8</sup> Wikipedia,  
<https://es.wikipedia.org/wiki/Implementaci%C3%B3n>,  
Consultado en Octubre 2016

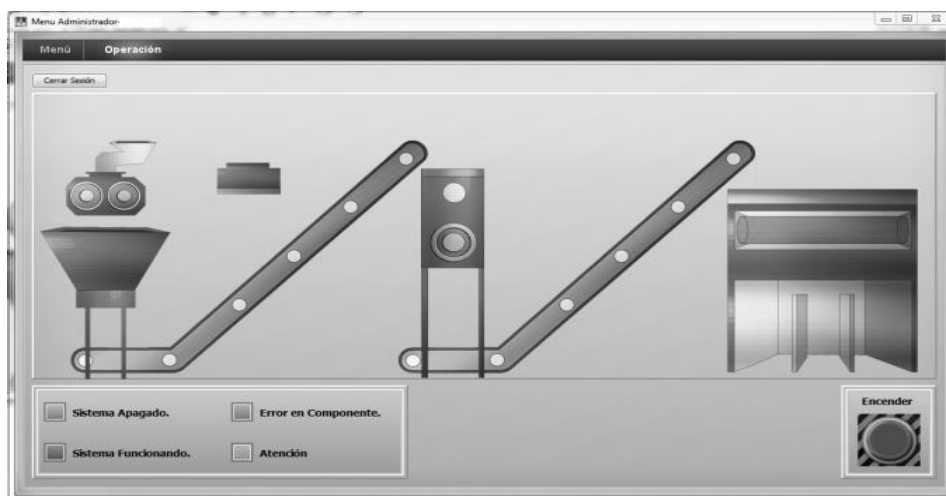


Fig. 6. Visualización de la maquinaria en reposo

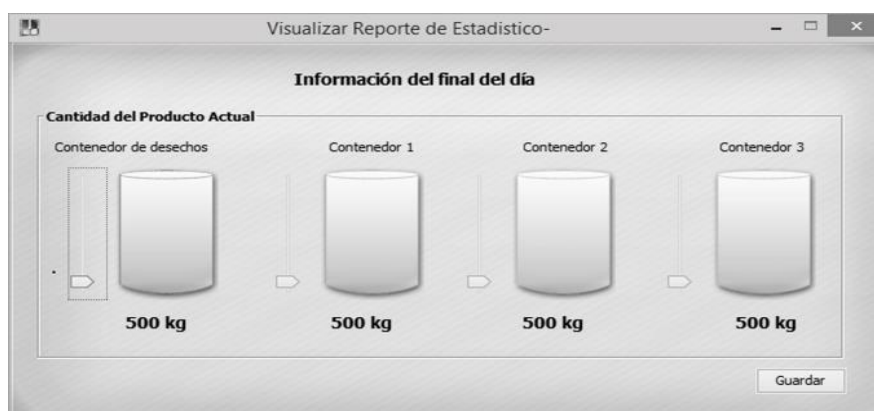


Fig. 7. Visualización de la cantidad de producto final por contenedores

modelo es una combinación de ambas definiciones, ya que se realizará la instalación del sistema en el equipo informático del cliente y al mismo tiempo se llevará acabo la evaluación final (o prueba beta) del producto [13], verificando el diseño, las especificaciones y el modelo propuesto para que todo funcionara como se especificó. Es necesario aclarar que las pruebas de los requerimientos fueron hechas en cada etapa siguiendo la metodología empleada.

La instalación de los dispositivos de control en la maquinaria es una parte elemental del proyecto para poder evaluar el software, no obstante, la elaboración de prototipos que simulen la

maquinaria física puede ser empleada, sin embargo, no sería lo óptimo. En la etapa de implementación es donde surgen los detalles no contemplados que están fuera del entorno de trabajo de los desarrolladores [12], lo cual alimenta / completa el producto final que será entregado al cliente.

Para realizar la implementación del sistema en la empresa se llevaron a cabo diversos procedimientos: 1) generar un instalador del SW, que desempaque e instale el software completo en el nuevo equipo , 2) probar la conectividad del equipo de control "PLC" con el sistema, 3) probar la comunicación con el PLC de forma bidireccional,

4) realizar pruebas generales de la funcionalidad del sistema con la maquinaria, 5) realizar prueba de los paros y los diversos tipos de fallas.

Concluidas las pruebas, se procede con la documentación 6) reportando todas las pruebas que fueron realizadas, 7) documentando de todos los fallos o errores en el software que fueron detectados y 8) se generó la documentación de los correctivos realizados.

Este proceso fue repetido cada vez que se aplicaban los correctivos necesarios para asegurar el correcto funcionamiento en la empresa. Por tal razón se generó una lista tipo *checklist* donde se verificaban los procesos realizados.

Este proceso de *checklist*, no solo se aplicó al final del proceso, en etapas previas a la entrega final (versiones finales de etapa) se definieron también listas tipo *checklist* donde se verificaban cada uno de los requerimientos y requisitos identificados y definidos por entregar y por la totalidad del sistema para verificar que todo haya sido probado y todo se haya validado, tratando de reducir al máximo los posibles errores no detectados.

## 4. Resultados

El seguimiento del modelo propuesto arrojó un total de 6 indicadores, los cuales se empelaron para validar el objetivo del sistema y en consecuencia la aprobación del modelo propuesto.

### 4.1. Resultados en cuanto al modelo de desarrollo

De manera general se requería validar al modelo y para ello era necesario llevarlo a cabo en al menos dos proyectos que incluyeran la generación de sistemas SCADA, uno de ellos fue el proyecto de Trituramiento y Pulverización del vidrio y el segundo fue el Lavado y Pulimiento del vidrio cribado. Ambos proyectos llegaron a buen término dentro de las fechas establecidas y cumplieron con la experiencia de usuario de los trabajadores. Situaciones que fueron obtenidas gracias a seguir una adecuada metodología de desarrollo de Software (RUP) y emplear un

lenguaje de modelado estándar internacional (UML).

### 4.2. Resultados en cuanto a la recolección de datos y presentación de información

La recolección de datos es invisible para los usuarios, sin embargo, es imprescindible para presentar información y tomar decisiones, los datos y procesos funcionales que fueron percibidos y vistos tanto por los directivos como por los operarios, haciendo alusión al buen control y automatización en tiempo real de la maquinaria conectada, a la buena visualización, recolección y procesamiento de los datos, generación de reportes, control de los elementos (avisos de mantenimiento de equipo o piezas en general) etc., es posible decir que se cumplieron con las especificaciones y objetivos de los proyectos, entregando adecuadamente ambos sistemas tipo SCADA con éxito. Muestra de ello, en la figura 6 se puede apreciar la pantalla de trabajo del proceso de trituramiento del vidrio del sistema SCADA generado.

De la misma manera se puede apreciar la recolección del producto terminado en contenedores separados obtenida en un día (ver figura 7).

### 4.3. Resultados respecto a gestión de información

El sistema fue creado para manejar la mayor cantidad de datos posibles y utilizarlos de manera eficiente: entre ellos se encuentra la gestión de la producción, la gestión de los fallos ocurridos, la gestión del tiempo efectivo de producción, así como del tiempo empleado para el mantenimiento y el tiempo en que se corrige una falla; la gestión de la maquinaria, la gestión del personal, etc.

Para poder hacer un análisis de toda la información generada el sistema tiene la capacidad de generar una serie de reportes de diferentes tipos, estos son generados en formato PDF y cualquiera de ellos se pueden generar diariamente y de forma automática o en formatos específicos por periodos largos, eligiendo el rango de fecha deseada, lo que da la posibilidad de generar reportes anuales, semestrales, trimestrales, etc.

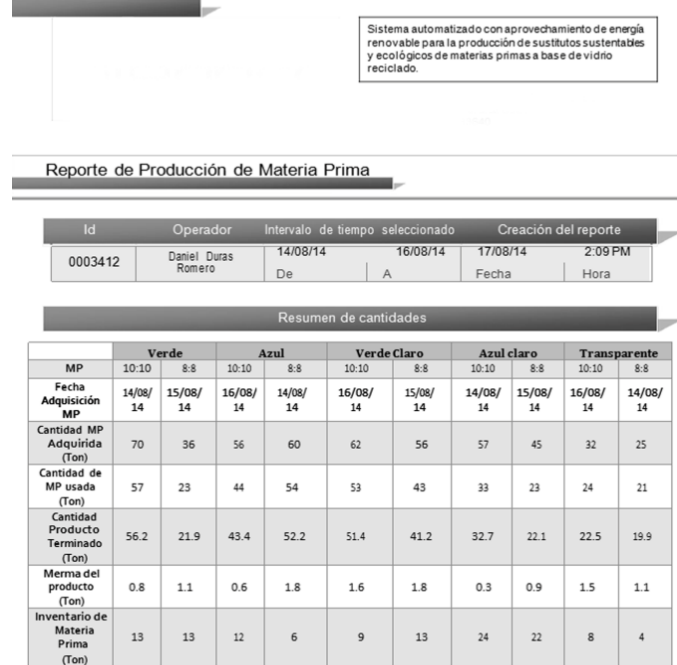


Fig. 8. Reporte de producción de materia prima (parte 1)

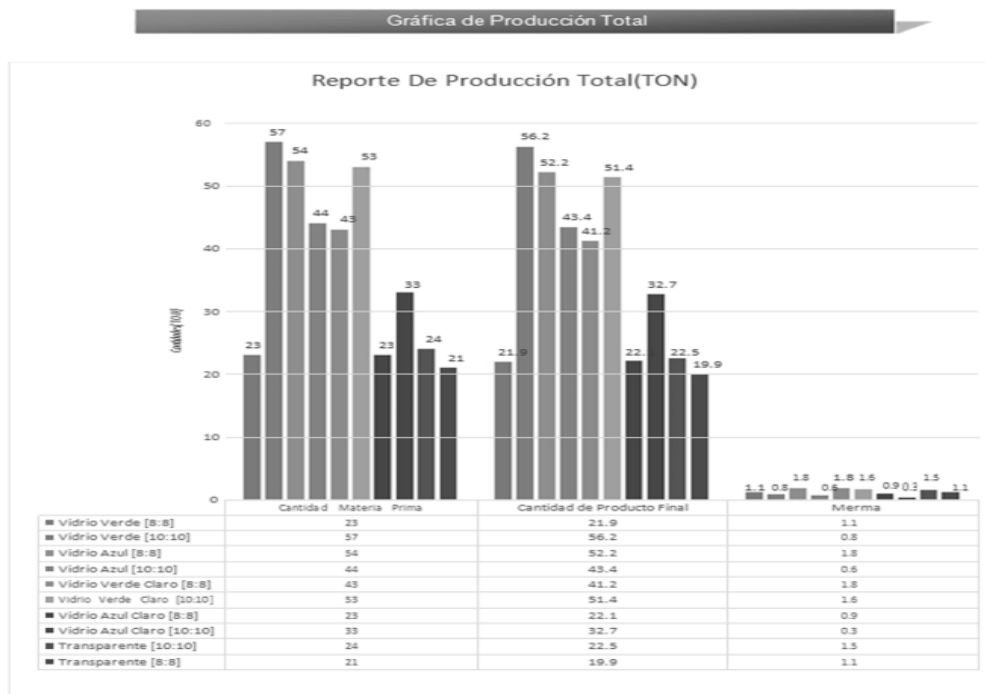


Fig. 9. Reporte de producción de materia prima (parte 2)

Un ejemplo para el reporte generado de producción de materia prima, se puede ver en las figuras 8 y 9 (parte 1 y 2).

#### **4.4. Resultados en cuanto a producción y control de producción**

El proceso original de la empresa se realizaba de una manera rustica, es decir, los procesos eran mayoritariamente manuales por lo que no se obtenían cantidades suficientes de producción. En su mejor día la empresa producía alrededor de 250 kg. Sin embargo, la utilización de una mejor maquinaria y la automatización del proceso de producción mejoró de forma considerable la producción, produciendo hasta 2.5 toneladas diarias, sin embargo, el proceso a como fue calculado podría producir en promedio un máximo 7.5 toneladas diarias.

El sistema fue creado para llevar el control de la producción por dos categorías, 1) basado en tamaño del vidrio triturado y 2) basado en el color del vidrio. Como es de saberse en un día se pueden producir diferentes tamaños y colores de vidrio y el sistema es capaz de registrar todos estos cambios para llevar un control diario basado en estas dos categorías.

#### **4.5. Resultados respecto a control de maquinaria**

El sistema tiene la posibilidad de llevar un control tanto de una maquinaria como de una parte específica de esta, es decir, se puede llevar el control de un motor, de un sensor, de un martillo, de una malla, etc. El control de esta forma sirve para que a la maquinaria se le proporcione un mantenimiento adecuado y siempre se encuentre en óptimas condiciones, con esta modalidad se puede saber cuándo se debe reemplazar una pieza o cuando se debe planear un paro de horas o de manera general para dar mantenimiento a una o varias máquinas o piezas.

#### **4.6. Resultados en cuanto a gestión de personal**

El sistema también cuenta con un registro del personal, es decir, se controla al equipo de trabajo, para cada proceso que se comience de

producción, se registra a los trabajadores que participan, con el objetivo de poder medir su eficiencia, es decir, saber si con un equipo de trabajadores ocurren más fallos o saber de forma específica quien no está trabajando correctamente para aplicar las medidas necesarias correspondientes.

Pudiendo también beneficiar a los empleados que más producen. De igual forma se registran las horas trabajadas por día, los accidentes que tiene cada empleado, la producción que obtiene cada equipo, etc.

### **5. Conclusiones**

En base a los resultados obtenidos en ambos proyectos y a la apreciación que se tuvo por parte de los directivos de la empresa, creemos que el modelo sugerido para la creación de un sistema SCADA de forma general es bueno y replicable.

La implementación de este tipo de sistemas en las plantas de producción, principalmente de nueva creación, deben contar con un sistema SCADA que ayude a controlar y a analizar sus datos de producción.

Previamente se comentó que, aquellas empresas que no toman en cuenta los datos empresariales, la tecnología y la innovación para obtener ventajas competitivas en su sector, son empresas que se están quedando obsoletas y cuentan con una gran probabilidad de desaparecer, hoy en día la flexibilidad que tengan las empresas en conjunto con la adaptación de la tecnología las hará más fuertes que nunca.

### **Agradecimientos**

Los autores agradecen el trabajo de todos los alumnos e investigadores que participaron en ambos proyectos y de todas aquellas personas que brindaron de su conocimiento para lograr el objetivo final. De igual forma se agradece el apoyo de Conacyt bajo el programa de Estímulos a la investigación, desarrollo tecnológico e innovación 2014 y 2015 (PROINNOVA 2014 y 2015) quien otorgó parte de los recursos empleados.

## Referencias

1. Bailey, D. & Wright, E. (2003). *Practical SCADA for Industry*. Newnes.
2. Bénabén, F. A. (2017). Information System agility to support collaborative organizations. *Enterprise Information Systems*, Vol. 11, No. 4, pp. 470–473. DOI:10.1080/17517575.2016.1269367.
3. Blanco-Llano, J. & Rodríguez-Hernández, A. (2011). Revisión, verificación y validación en un proceso de desarrollo de software. *Ingeniería industrial*, Vol. 32, No. 1, pp. 28–36.
4. Cañavate, A. M. (2003). Hipertext.net. Recuperado el 2016, de [https://www.upf.edu/hipertextnet/numero-1/sistem\\_infor.html](https://www.upf.edu/hipertextnet/numero-1/sistem_infor.html).
5. Castañeda, S. S., Ramírez, D. C., & Mayorga, V. M. (2014). El Enterprise Risk Management (ERM) para la evaluación de riesgos estratégicos en microempresas comerciales hidalguenses. *Acta Universitaria*, Vol. 24, No. NE-1, pp. 95–104. DOI: 10.15174/AU.2014.713.
6. Castells, M. (1997). La economía información y el proceso de globalización en la era de la información. *Economía, Sociedad y Cultura*, Vol. 4, No. 1, pp. 93–178.
7. Chen, Y. Y. & Huang, H. L. (2012). Knowledge management fit and its implications for business performance: A profile deviation analysis. *Knowledge-Based Systems*, Vol. 27, pp. 262–270. DOI: 10.1016/j.knosys.2011.11.012.
8. Clements, P. & Northrop, L. (2001). *Software Product Lines: Practices and Patterns*. Addison-Wesley.
9. Cornella, A. (2000). *Infonomía.com. La empresa es información*. Bilbao: Deusto S.A. ediciones.
10. Cortés-Verdín, M. K., López-Araujo, M. L., & Ocharán-Hernández, J. O. (2012). Development of a plug-in for Variability Modeling in Software Product Lines. *Acta Universitaria*, Vol. 22, No. NE-1, pp. 134–140.
11. Farzin, M. R., Kahreh, M. S., Hesani, M., & Khalouei, A. (2014). A survey of Critical Success Factors for Strategic Knowledge Management Implementation: Applications for Survey Sector. *Procedia-Social and Behavioral Sciences*, Vol. 109, pp. 595–599. DOI: 10.1016/j.sbspro.2013.12.512.
12. Fifarek, A. (2008). To Beta Test, or Not to Beta Test: Is That the Question?. *Library Hi Tech News*, Vol. 25, No. 1, pp. 30–31. DOI: 10.1108/07419050810877544.
13. Gaikwad, H. & Pillai, S. (2014). Testing and Evaluation: A case study Approach. *International Proceedings of Economics Development and Research*, Vol. 75, pp. 24–28.
14. Guerrero-García, J., González-Calleros, J. M., & Vanderdonckt, J. (2012). Workflow User Interfaces Patterns. *Acta Universitaria*, Vol. 22, No. NE-1, pp. 62–68. DOI: 10.15174/au.2012.343.
15. Laudon, K. C. & Laudon, J. P. (2012). *Sistemas de Información Gerencial*. Pearson Education.
16. Letelier-Torres, P. & Sanchez-López, E. A. (2003). Metodologías ágiles en el Desarrollo de Software. *VIII Jornadas de Ingeniería del Software y Bases de Datos*, pp. 1–8.
17. Nikolic, A. & Drexler, J. (2015). *Seguridad informática para sistemas de control industrial en instalaciones nucleares*. San Carlos de Bariloche.
18. Park, C., Bajimaya, S., Park, S., Wang, G., Kwak, J., & Han, K. (2007). Development of Virtual Simulator for Visual Validation of PLC Program. *International Conference on Computational Intelligence for Modeling Control and Automation*, pp. 32–36. DOI: 10.1109/CIMCA.2006.86.
19. Penin, A. R. (2007). *Sistemas Scada*. España: marcombo.
20. Pérez-López, E. (2015). Los sistemas SCADA en la automatización industrial. *Tecnología en Marcha*, Vol. 28, No. 4, pp. 3–14.
21. Piedrafita, M. R. (2004). *Ingeniería de la automatización industrial*. Madrid: Ra-Ma.
22. Pressman, R. (2001). *Ingeniería del software: Un enfoque práctico*. McGraw-Hill.
23. Sánchez, S., Sicilia, M. Á., & Rodríguez, D. (2012). *Ingeniería del Software - un enfoque desde la guía SWEBOK*. Madrid: Alfaomega.
24. Sarduy-Gómez, R. J., Guzmán del Río, D., & Reyes Calvo, R. (2008). *Temas especiales de instrumentación y control*. La Habana, Cuba: Felix Varela.
25. Senn, J. (1993). *Análisis y diseño de sistemas de información*. McGraw-Hill.
26. Serna, M. E., & Martínez, M. R., & Tamayo, O. P. (2017). Un modelo para determinar la madurez de la automatización de las pruebas del software como área de investigación y desarrollo. *Computación y Sistemas*, Vol. 21, No. 2, pp. 337–352. DOI: 10.13053/CyS-21-2-2723.
27. Sol, M. R. (2008). *Diseño e implantación de un sistema SCADA para una planta de producción y envasado de líquidos*. Barcelona: UAB.
28. Sommerville, I. (2011). *Ingeniería de Software*. Pearson Educación.
29. Stouffer, K., Falco, J., & Scarfone, K. (2015). *Guide to supervisory control and data acquisition*



(SCADA) and industrial control systems security, *Special Publication NIST-SP-800-82-Rev.2-2015*. Gaithersburg: National Institute of Standards and Technology.

30. Van-Gurp, J., Svahnberg, M., & Bosch, J. (2001). On the Notion of Variability in Software Product Lines. *Proc. Of the Working IEEE/IFIP Conference on Software Architecture (WICSA '01)*, pp. 45–54. DOI: 10.1109/WICSA.2001.948406.
31. Yehia, S. (2016). Journey to the center of the plant: SCADA systems information flow. *Control Engineering*, Vol. 63, No. 1.
32. Zaeh, M., Ehrenstrasser, M., Poernbacher, C., & Wuensch, G. (2003). Emerging Virtual Machine Tools. *Design Automation Conference*, Vol. 2, pp. 513–518.
33. Zaeh, M., Poernbacher, C., & Milberg, J. (2005). A Model-Based Method to Develop PLC Software for Machine Tools. *CIRP Annals - Manufacturing Technology*, Vol. 54, No. 1, pp. 371–374. DOI: 10.1016/S0007-8506(07)60125-5.

Article received on 11/03/2017; accepted on 07/05/2018.  
Corresponding author is José Luis Ochoa-Hernández.