



UNIVERSIDAD
DE PIURA

REPOSITORIO INSTITUCIONAL
PIRHUA

DESARROLLO DE UN SISTEMA SCADA PARA USO EN PEQUEÑAS Y MEDIANAS EMPRESAS

David Aguirre Zapata

Piura, abril de 2013

Universidad de Piura

Área Departamental de Ingeniería Industrial y de Sistemas

Aguirre, D. (2013). *Desarrollo de un sistema SCADA para uso en pequeñas y medianas empresas*. Tesis de pregrado en Ingeniería Industrial y de Sistemas. Universidad de Piura. Facultad de Ingeniería. Programa Académico de Ingeniería Industrial y de Sistemas. Piura, Perú.



Esta obra está bajo una [licencia](#)
[Creative Commons Atribución-](#)
[NoComercial-SinDerivadas 2.5 Perú](#)

Repositorio institucional PIRHUA – Universidad de Piura

UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERIA



“Desarrollo de un sistema SCADA para uso en pequeñas y medianas empresas”

Tesis para optar el Título de
Ingeniero Industrial y de Sistemas

David Aguirre Zapata

Asesores: William Ipanaqué Alama – Omar Hurtado Jara

Piura, Abril 2013

Prólogo

En el año 2008 se desarrolló un proyecto denominado: “Desarrollo e investigación de embebidos (TIC) para aplicación de automatización y control de bajo costo para pequeñas y medianas industrias”, proyecto financiado por el FINCyT (Unidad coordinadora del programa de ciencia y tecnología) y desarrollado por la Universidad de Piura. Dicho proyecto tenía varios componentes, entre ellos: El desarrollo de tarjetas programables, software de bajo nivel y software de alto nivel, entre ellos un sistema SCADA.

El desarrollo del proyecto llevó más de 2 años de investigación, pruebas e informes en los distintos niveles de la organización del proyecto. Al final del mismo se obtuvieron resultados satisfactorios, tanto fue el éxito que algunos de los subproductos fueron comprados para su uso en determinadas empresas.

Con esto nació la idea de seguir desarrollando el software y mejorarlo, pues en esa versión el software tenía sus limitaciones, faltaba más investigación. Posteriormente al proyecto se desarrolló el denominado sistema “Soft-Control” teniendo como autor el mismo de la presente tesis. “Soft-Control” es una versión mejorada del sistema SCADA implementado en ese proyecto.

Resumen

El trabajo que presentaré a continuación muestra el desarrollo de un sistema automatizado para ser implementado en empresas de bajos recursos, las cuales no pueden acceder a sistemas comerciales similares. Dichas empresas siguen manteniendo elevados costos en registros a mano, impresiones, correos y mano de obra, consecuentemente no pueden competir con las empresas extranjeras que si tienen fondos para invertir en ese tipo de software.

El desarrollo de dicho software promueve el uso de software libre, respetando la complejidad, compatibilidad, recursos y los requerimientos de operatividad y seguridad de pymes.

El desarrollo de la tesis se divide en 4 partes: marco teórico, diseño, seguridad, e implementación. Esos 4 capítulos engloban toda la información y los procedimientos con que fue desarrollado "Soft-Control". Una vez visto todos los capítulos, el lector será capaz de entender términos clave en automatización, obtendrá conocimientos informáticos, conocerá más acerca de lenguajes de programación tanto web como de escritorio, conocerá el concepto de base de datos, además sabrá los pasos para realizar un diseño de software según la ingeniería de software, estará al tanto de las tecnologías de software libre que existen hoy en día y podrá proponer una solución empresarial para la automatización industrial con un software de costo muy bajo, eficiente y confiable.

Índice

Introducción	1
Capítulo 1 Marco Teórico	3
1.1 Historia.....	3
1.2 Sistemas SCADA	4
1.2.1 Ventajas.....	5
1.2.2 Funciones:	6
1.3 Tipos de sistemas SCADA.....	7
1.3.1 Sistemas SCADA abiertos y propietarios	7
1.3.2 Sistemas SCADA comerciales y gratuitos	8
1.4 Partes de un sistema SCADA.....	8
1.4.1 Unidades Terminales Maestras	9
1.4.2 Unidades Terminales Remotas.....	9
1.5 Sistemas SCADA Comerciales	9
1.5.1 Sistemas SCADA comerciales	9
1.5.2 Sistemas SCADA Open Source de distribución gratuita	10
1.6 Criterios de elección.....	11
1.7 Lenguaje Java.....	12
1.8 PHP	13
1.9 Modelo Cliente - Servidor:.....	14
1.9.1 Definición:.....	14
1.9.2 Ventajas del modelo:.....	15
1.10 Servidor Web	15
1.11 Base de Datos.....	17
1.12 Comunicaciones	18

1.12.1 Protocolo Ethernet.....	18
1.12.2 Protocolo Serial:.....	19
1.12.3 Comunicación OPC:.....	20
1.12.3.1 Elementos de la comunicación OPC:.....	21
Capítulo 2: Diseño.....	22
2.1 Alcance del software.....	22
2.2 Perspectiva del producto.....	23
2.3 Capacidades Generales.....	23
2.4 Restricciones generales.....	23
2.5 Entorno operacional.....	24
2.6 Características de los Usuarios.....	24
2.7 Modelo de Casos de Uso.....	24
2.8 Especificación Textual.....	25
2.9 Modelo Conceptual.....	28
2.10 Modelo relacional de la base de datos del sistema.....	29
2.10.1 Diccionario de datos de las tablas.....	29
2.10.2 Detalle del modelo relacional del Sistema:.....	31
2.11 Arquitectura Lógica.....	32
2.11.1 Especificación de las capas.....	32
2.11.2 Diagrama de Componentes.....	32
2.12 Arquitectura Física.....	33
2.12.1 Especificación de las herramientas tecnológicas y herramientas a utilizar.....	33
2.12.2 Diagrama de Despliegue.....	34
2.13 Diseño de la Interfaz.....	35
2.14 Diseño de la Lógica de Aplicación o de Negocio.....	53
2.14.1 Diagrama de Clases de Diseño.....	53
2.14.2 Detalle del diagrama de clase del diseño.....	54
Capítulo 3: Implementación.....	64
3.1 Metodología de desarrollo:.....	64
3.1.1 Metodologías actuales.....	64
3.1.2 Selección de la metodología.....	65
3.2 Librerías Usadas.....	66
3.3 Seguridad.....	68
3.3.1 Encriptación MD5.....	68

3.3.2 Encriptación MySQL	69
3.3.3 Protocolo de comunicación propio.....	69
Conclusiones	70
Glosario	71
Bibliografía	72
ANEXOS.....	75

Introducción

Hoy en día existen muchos sistemas sofisticados que permiten el control de las plantas industriales, que facilitan el acceso a los datos y generan información vital para la empresa. Muchos de ellos con un costo tan alto que obligan a las pequeñas empresas a renunciar al intento de implantar un nuevo sistema que les permita progresar. Es por ello que en el siguiente documento se plantea diseñar e implementar un software de supervisión, control y adquisición de datos (SCADA) eficiente y que además sea lo suficientemente accesible para las empresas que no disponen de los recursos suficientes para adquirir uno en el mercado actual.

El sistema SCADA es una aplicación de software especialmente diseñada para funcionar sobre computadores, generalmente en la parte de control de producción, proporcionando comunicación con los dispositivos de campo como bombas, motores, etc. y controlando el proceso de forma automática desde pantallas de ordenador a las cuales llamaremos HMI (Interfaz Humano Máquina). También provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros usuarios supervisores dentro de la empresa (supervisión, control calidad, control de producción, almacenamiento de datos, etc.).

Dichos sistemas se caracterizan por el uso del denominado *feedback* (retroalimentación), proceso de compartir observaciones, preocupaciones y sugerencias, con la intención de recabar información para mejorar o modificar diversos aspectos del funcionamiento de una organización. La realimentación es la forma en la cual el sistema contrasta los datos anteriores con los actuales, de esta forma podemos inferir si el proceso sigue un comportamiento normal o es afectado por un factor.

En la actualidad el software de código abierto está en una etapa de fuerte crecimiento y ya está compitiendo al mismo nivel que el software pagado, incluso los sistemas operativos que usan las empresas para su operación son de código abierto pues ofrecen mayor seguridad que los comerciales. Otra característica del software de pago es que está diseñado para trabajar sobre plataformas de pago, las cuales piden muchos recursos, tienen muchas restricciones de implementación y no son totalmente integrables con la empresa, además de tener también un costo elevado.

Debido a las razones anteriormente mencionadas, en los siguientes capítulos veremos el diseño e implementación de un sistema SCADA usando diferentes tecnologías de bajo costo y de fácil implementación para una empresa que desea salir adelante apostando por tecnologías de código abierto eficiente, compatible y confiable.

Capítulo 1

Marco Teórico

1.1 Historia

Para empezar a hablar de los sistemas SCADA, tenemos que conocer la historia detrás de ellos. Es necesario hablar sobre el proceso evolutivo, el cual sigue una serie de hitos, los cuales uno tras otro permitieron evolucionar la tecnología disponible en ese entonces hasta la que disponemos en el mundo actual, algunos de los acontecimientos más importantes fueron:

- Hacia el 2000 a.c existía el ábaco el cual se considera la primera máquina para realizar operaciones de cálculo.
- En 1614 John Napier da a conocer los logaritmos o números artificiales, gracias a ellos las multiplicaciones se convierten en sumas, las potencias en productos, y las raíces en divisiones, simplificando enormemente los cálculos matemáticos.
- Leonardo Da Vinci entre 1452 y 1519 esbozó la primera máquina capaz de realizar operaciones matemáticas.
- En 1642 Blaise Pascal se inventa la “Pascalina”, considerada la primera calculadora.
- Charles Babbage, tras inventar la máquina llamada “locura de Babbage” debido a las enormes dificultades de su puesta en marcha y de la complejidad de la misma, daba el siguiente paso en el cálculo analítico, sin embargo, su prototipo no pudo ser llevado a la práctica pues en su tiempo no existía la tecnología para realizarla.
- En los años 70 aparece la primera calculadora electrónica.
- En los 80 el sistema operativo Windows.
- Posteriormente se dio el desarrollo de múltiples aplicaciones para el uso industrial, una de ellas fue el sistema SCADA, un software que revolucionaría la industria y la automatización (García, 2012).

Por otro lado, en la industria de la automatización la historia empieza con el término DCS (*Distributed control system* o Sistema de control distribuido). Los DCS son sistemas que cumplen funciones de control a través de una serie de módulos de control automáticos independientes y distribuidos en la planta.

La estrategia de la arquitectura de los DCS es evitar que el control total de la planta esté centralizado en una sola unidad. Los DCS se desarrollaron a base de dispositivos o PLC (*Programmable logic controller* o controlador lógico programable). Las decisiones son enviadas a los actuadores (Mecanismos que modifican variables en el proceso), los cuales hacen modificaciones para mantener el proceso estable.

Sin embargo, la tendencia ahora es integrar los sistemas SCADA a los DCS, es decir orientar la parte de supervisión a los sistemas SCADA, pero usando los DCS como sistemas independientes que mandan señales al sistema SCADA. (Corrales, 2007)

1.2 Sistemas SCADA

Su nombre proviene de las siglas "*Supervisory Control And Data Acquisition*" (Sistema de control, supervisión y adquisición de datos). Es un sistema basado en computadores que permite supervisar y controlar variables de proceso a distancia, proporcionando comunicación con los dispositivos de campo (controladores autónomos) y controlando el proceso de forma automática por medio de un software especializado. También provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros usuarios supervisores dentro de la empresa (supervisión, control calidad, control de producción, almacenamiento de datos, etc.) (Lozano, 2012)

Los sistemas SCADA involucran muchos subsistemas, por ejemplo, la adquisición de los datos puede estar a cargo de un PLC (Controlador Lógico Programable) o de dispositivos los cuales toman las señales y las envía a las estaciones remotas usando un protocolo determinado, otra forma podría ser que una computadora realice la adquisición vía un hardware especializado y luego esa información la transmita hacia un equipo de radio vía su puerto serial, y así existen muchas otras alternativas.

Las tareas de Supervisión y Control generalmente están más relacionadas con el software SCADA, en él, el operador puede visualizar en la pantalla del computador de cada una de las estaciones remotas que conforman el sistema, los estados de ésta, las situaciones de alarma y tomar acciones físicas sobre algún equipo lejano, la comunicación se realiza mediante buses especiales o redes LAN. Todo esto se ejecuta normalmente en tiempo real, y están diseñados para dar al operador de planta la posibilidad de supervisar y controlar dichos procesos.

El sistema SCADA actúa sobre los dispositivos instalados en la planta, como son los controladores, autómatas, sensores, actuadores, registradores, etc. Además permite

controlar el proceso desde una estación remota, para ello el software brinda una interfaz gráfica que muestra el comportamiento del proceso en tiempo real. Generalmente se vincula el software al uso de una computadora o de un PLC, la acción de control es realizada por los controladores de campo, pero la comunicación del sistema con el operador es necesariamente vía computadora. Sin embargo el operador puede gobernar el proceso en un momento dado si es necesario.

1.2.1 Ventajas

Entre las principales ventajas de un sistema SCADA podemos destacar:

- **Economía:** Es más fácil ver qué ocurre en la instalación desde la oficina que enviar a un operario a realizar la tarea de supervisión. Ciertas revisiones se convertirán en innecesarias y con eso se estaría ahorrando un monto importante en mano de obra.
- **Accesibilidad:** Tener toda la información al alcance de un clic, podemos modificar el nivel de funcionamientos de los dispositivos de planta, apagar y encender máquinas, consultar el estado de las mismas en el tiempo y tomar una decisión en tiempo real o incluso antes de un suceso, todo eso de manera remota y a cualquier hora.
- **Mantenimiento:** La adquisición de datos y su supervisión da la posibilidad de presentarlos de manera inteligible para usuarios no especializados y de avisar cuando se aproximen las fechas de revisión o cuando una máquina tenga más fallos de los considerados normales.
- **Ergonomía:** La relación entre el usuario y el proceso es menos tediosa. Los modernos ordenadores, con sus prestaciones gráficas, sustituyen a los grandes paneles, repletos de cables, pilotos y del excesivo papeleo informativo.
- **Gestión:** Todos los datos recopilados pueden ser valorados de múltiples maneras mediante herramientas estadísticas, gráficas, valores tabulados, etc., que permitan explotar el sistema con el mejor rendimiento posible.
- **Flexibilidad:** Cualquier modificación de alguna de las características del sistema de visualización no significa un gasto en tiempo y medios, pues no hay modificaciones físicas que requieran la instalación de un cableado o del contador.
- **Conectividad:** Se buscan sistemas abiertos, es decir, sin secretos ni sorpresas para el integrador. La documentación de los protocolos de comunicación actuales permite la interconexión de sistemas de diferentes proveedores y evita la existencia de

lagunas informativas que puedan causar fallos en el funcionamiento o en la seguridad. (García, 2012)

1.2.2 Funciones:

Las funciones básicas de un sistema SCADA son las que se describen a continuación:

- Supervisión remota de instalaciones: Mediante esta función, el usuario es capaz de conocer el estado de las instalaciones bajo su responsabilidad y coordinar eficientemente las labores de producción y mantenimiento en el campo. El intervalo de recolección periódica de la información del campo depende de las dimensiones, pero generalmente está en el orden de unos cuantos milisegundos.
- Control remoto de instalaciones: Los sistemas SCADA permiten activar o desactivar equipos remotos de manera automática o a solicitud del operador. Igualmente es posible realizar ajustes remotamente a parámetros del proceso.
- Procesamiento de información: En algunos casos, los datos capturados requieren procesamiento adicional, a efectos de consolidar información proveniente de diferentes lugares remotos, como por ejemplo de balance de masa entre diferentes instalaciones.
- Presentación de gráficos dinámicos: Esto se refiere al despliegue de pantallas con el diagrama del proceso conteniendo información instantánea del comportamiento del mismo.
- Generación de reportes: Los sistemas SCADA permiten la generación automática o a petición de reportes impresos de producción y balances.
- Presentación de alarmas: Mediante esta función se alerta al operador sobre la ocurrencia de condiciones anormales o eventos que pudieran requerir su intervención. Normalmente, la criticidad del evento o alarma se indica mediante el uso de colores y/o señales auditivas. Las alarmas se registran para análisis posteriores.
- Almacenamiento de información histórica: Los sistemas SCADA permiten registrar y almacenar información operacional y alarmas. Por ejemplo, se pueden llevar datos de los últimos 5 minutos, 1 hora, 1 día, 1 mes y hasta un año.

- Presentación de gráficos de tendencias: Con información en tiempo real o histórico, se pueden construir gráficos e inferir el comportamiento de variables operacionales en el tiempo.
- Programación de eventos: Se refiere a la posibilidad de programar en el tiempo la generación de reportes, despliegue de diagramas del proceso o activación de tareas o comandos del sistema. (Geocities, 2012)

Otras prestaciones que algunos SCADA ofrecen son las siguientes:

- Posibilidad de crear paneles de alarma, que exigen la presencia del operador para reconocer una parada o situación de alarma, con registro de incidencias.
- Generación de datos históricos de las señales de planta, que pueden ser volcados para su proceso sobre una hoja de cálculo.
- Ejecución de programas, que modifican la ley de control, o incluso anular o modificar las tareas asociadas al autómatas, bajo ciertas condiciones.
- Posibilidad de programación numérica, que permite realizar cálculos aritméticos de elevada resolución sobre la CPU del ordenador. (Diaz, 2012)

1.3 Tipos de sistemas SCADA

Los distintos SCADA pueden dividirse en dos categorías y según estas definirlos:

1.3.1 Sistemas SCADA abiertos y propietarios

- Los sistemas abiertos u Open son aquellos desarrollados para poder ser aplicados a cualquier tipo de tecnología o dispositivo de control, es decir si se necesita enlazar un equipo de distintos fabricantes, es necesario solo contar con los drivers que interpreten los distintos códigos de comunicación utilizados. La principal ventaja de este tipo de sistema es su capacidad de crecimiento conjunto con la planta, es decir nuevos equipos pueden ser implementados así sean de distintos fabricantes.
- Los sistemas propietarios son aquellos desarrollados por los propios fabricantes de equipos o dispositivos de control, los cuales se comunican entre sí con sus propios drivers; la principal desventaja de este tipo de software SCADA es la gran dependencia que se tiene del proveedor del sistema.

1.3.2 Sistemas SCADA comerciales y gratuitos

- Un sistema SCADA comercial es aquel en el que por lo general su desarrollo está a cargo de una compañía, la cual se encarga de crear todas las interfaces necesarias para comunicar los distintos dispositivos, y una vez finalizado esto, entregar al usuario un producto de fácil uso. Mientras más confiable y amigable sea el software, este es más costoso, por tanto de difícil acceso para pequeñas empresas las cuales se ven obligadas a contar con todo un personal a disposición del monitoreo de la planta.
- Un sistema SCADA gratuito por lo general fue creado como un SCADA comercial, con el transcurso del tiempo se vio que había mayores ventajas en poner estos sistemas con su código de programación en forma abierta a disposición de distintos desarrolladores alrededor del mundo, los cuales cooperan con su desarrollo, por lo general la única condición para poder adquirir estos software es comprometerse a que una vez logrado el objetivo buscado, este conocimiento sea compartido (Udep, 2008).

1.4 Partes de un sistema SCADA

Un sistema SCADA ha sido creado para monitorear desde una sola central, los distintos procesos de una o varias plantas, un esquema regular de un sistema SCADA, consta de una Unidad Terminal Maestra (MTU), de Unidades Terminales Remotas (RTU) y los medios físicos de comunicación, tal y como se muestra en la Figura 1.

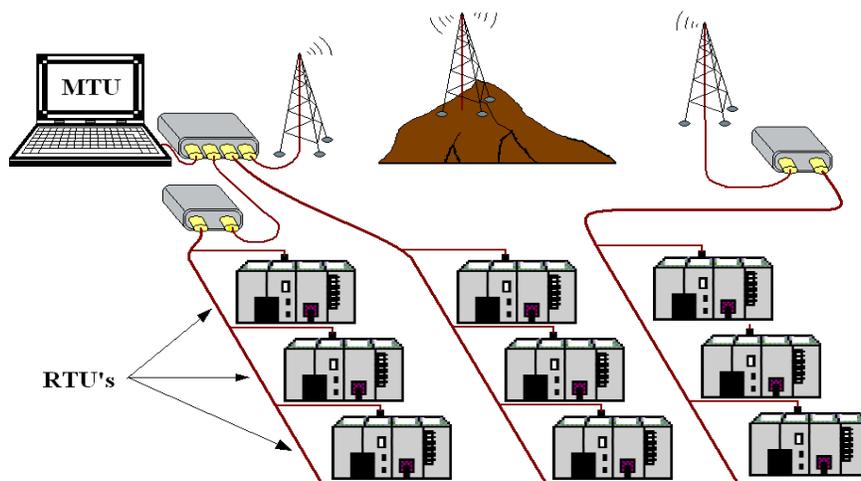


Figura 1 - Componentes del sistema SCADA (Udep, 2008)

1.4.1 Unidades Terminales Maestras

Es el centro del sistema, es el componente del cual el personal de operaciones se valdrá para visualizar las distintas variables de los procesos en la planta; generalmente una MTU es un computador PC, de regular capacidad, el cual cumple funciones no solo de monitoreo, sino a la vez de almacenamiento y procesado ordenado de datos, los cuales servirán para las distintas aplicaciones que el operario o usuario requiera.

1.4.2 Unidades Terminales Remotas

Una RTU es unidad de control y adquisición de datos independiente, por lo general ha estado basada en microprocesadores, los cuales monitorean y controlan equipo en algunas ubicaciones remotas desde la estación central. Esta tarea primaria es controlar y adquirir datos desde los equipos de los procesos en la ubicación remota y transferir esta información hacia la estación central. La configuración de cómo descargar dinámicamente esta información, es mediante una configuración dada por la estación central. (Udep, 2008)

1.5 Sistemas SCADA Comerciales

Se ha realizado una cotización de distintos sistemas SCADA existentes en el mercado, lográndose comprobar su elevado costo, lo que las hace inaccesibles para las empresas en desarrollo.

1.5.1 Sistemas SCADA comerciales

La Tabla 1 muestra una lista detallada los principales SCADAS comerciales, sus proveedores y su costo.

Tabla 1 - Lista de SCADAS comerciales (Udep, 2010)

Software SCADA Comerciales			
Nombre	Compañía	Tipo(abierto/Propietario)	Costo
JAKO SCADA	JANUS (Arg.)	abierto	
Superview	Novus Automation(Bra.)	abierto	400\$
Proficy® HMI/CIMPLICITY HMI	GE Fanuc	abierto	
Proficy® HMI/iFIX	GE Fanuc	abierto	
Invesys Scada System	Foxboro	propietario	1900\$
Simatic WinCC	Siemens	propietario	2600£
DAQ Factory	Azeo Tech	abierto	2500\$
LookOut	National Instrument	abierto	1045\$
WinLog PRO	SIELCO SISTEMI	abierto	400£
WinLog Lite	SIELCO SISTEMI	abierto	49£
Intouch HMI	Wonderware	abierto & propietario	3100\$
CitecSCADA	Schneider Electric	propietario	
ClearSCADA	CONTROL MICROSYSTEM	abierto & propietario	

Cabe resaltar que “Soft-Control” al estar basado en tecnologías de código libre, no presenta ningún tipo de costo. El precio de venta que se propone en base al tiempo de desarrollo del software sería 1000 dólares anuales por nodo.

1.5.2 Sistemas SCADA Open Source de distribución gratuita

La Tabla 2 muestra una lista detallada los principales SCADAS comerciales, sus proveedores y algunos datos de funcionamiento.

Tabla 2 - Algunos SCADAS libres (Udep, 2010)

Softwares SCADA OPEN SOURCE de distribución gratuita				
<i>Nombre</i>	<i>Desarrolladores</i>	<i>Plataforma</i>	<i>Base de datos</i>	<i>Interfaces de comunicación</i>
EPICS	Argonne - National Laboratory	Linux, MS Windows, MacOS and RTEMS	Propio	MODBUS, UDP, Ethernet
LINKIDOY	Axarquia - Andalucía - España	Linux - Python	MySQL	FTP, SFTP, MODBUS, WEB, UDP
PVBROWSER	Web	Linux, MS Windows	MySQL	MODBUS, UDP, Ethernet, WEB

La tabla 3 nos muestra otros sistemas SCADA en el mercado:

Tabla 3 - Otros sistemas SCADA (Lozano, 2012)

Nombre	Proveedor
Aimax	Desin Instruments S.A.
CUBE	Orsi España S.A.
FIX	Intellution
Monitor Pro	Schneider Electric.
SCADA InTouch	LOGITEK
SYSMAC SCS	Omron
Scatt Graph 5000	ABB
VB ScadaLadder	Microsoft
addVANTAGE Pro	ADCON Telemetry
LabView	National Instruments

1.6 Criterios de elección

Un sistema de control cualquiera es útil, evidentemente, mientras funcione correctamente. En caso contrario puede crear problemas de forma directa (mal funcionamiento de un sistema de potabilización de agua), o indirecta (el fallo del control sobre una estación transformadora puede hacer que el sistema de control central provoque un efecto dominó al sobrecargar las estaciones adyacentes, que no están preparadas para ello).

La reacción de un sistema ante situaciones inesperadas determinará su grado de fiabilidad, es decir, el tiempo de operación del mismo, y puede mejorarse mediante el uso de técnicas de diseño adecuadas.

Los parámetros que influyen en la decisión de desarrollo o de compra pueden ser los siguientes:

1. Disponibilidad

Por disponibilidad de un sistema se entiende la medida en la que sus parámetros de funcionamiento o partes se encuentran listas o en espera para el funcionamiento del mismo. Estamos hablando de tener instalados y en reserva tanto partes de hardware como de software.

2. Robustez

Ante un fallo de diseño, un accidente o una intrusión, un sistema eficiente debe poder mantener un nivel de operatividad suficiente como para mantener unos mínimos de servicio. Si las cosas empiezan a ir mal, ¿cuánto aguantará el sistema antes de empezar a fallar? Es lo que llamaríamos el plan de contingencia.

3. Seguridad

Un fallo en el diseño, un usuario malintencionado o una situación imprevista podrían alterar los parámetros de funcionamiento de un sistema. Hoy en día cualquier sistema de control puede utilizar uno o varios métodos de comunicación para enlazar todos los puntos de control de un proceso y, en el momento en que se utilizan sistemas de comunicación que implican el acceso desde múltiples puntos, no siempre dentro de la empresa, es posible que alguno de estos accesos sea no deseado.

4. Prestaciones

El equipo debe poder asimilar toda la información incluso bajo condiciones extremas de trabajo de manera que no se pierda ningún dato aunque no necesariamente los almacene en tiempo real.

5. Mantenibilidad

Los tiempos de mantenimiento pueden reducirse al mínimo si el sistema está provisto de buenas herramientas de diagnóstico para realizar tareas de mantenimiento preventivo, modificaciones y pruebas simultáneas al funcionamiento normal del sistema.

6. Escalabilidad

Es importante que el sistema pueda ampliarse con nuevos componentes tanto de software como de hardware según los requerimientos de la operación. (Rodriguez Penin, 2012)

1.7 Lenguaje Java

“Soft-Control” fue desarrollado principalmente en el lenguaje de programación Java, debido a que este lenguaje es el líder de los lenguajes de programación para el software libre, dispone de librerías de código abierto útiles y confiables, además de tener un costo muy bajo es muy rápido y sus programas pueden ejecutarse en diferentes sistemas operativos, lo que nos da la posibilidad de que la empresa al adquirir un sistema operativo de código abierto pueda trabajar con el sistema sin ningún problema.

Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión. (Java, 2013)

Desde 1995 entonces, Sun Microsystems ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del “*Java Community Process*”, otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Con la programación en Java, se pueden realizar distintos aplicativos, como son applets, que son aplicaciones especiales, que se ejecutan dentro de un navegador al ser cargada una página HTML en un servidor WEB. Por lo general los applets son programas pequeños y de propósitos específicos.

Otra de las utilidades de la programación en Java es el desarrollo de aplicaciones, que son programas que se ejecutan en forma independiente, es decir con la programación Java, se pueden realizar aplicaciones como un procesador de palabras, una hoja que sirva para cálculos, una aplicación gráfica, etc. En resumen cualquier tipo de aplicación se puede realizar con ella. Java permite la modularidad por lo que se pueden hacer rutinas individuales que sean usadas por más de una aplicación, por ejemplo tenemos una rutina de impresión que puede servir para el procesador de palabras, como para la hoja de cálculo.

La programación en Java, permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar. (Lenguajes de Programación, 2012)

1.8 PHP

Una parte de “Soft-Control” fue desarrollada en el lenguaje de programación web PHP, el cual es también un lenguaje de programación de código abierto, es decir es un software libre. La parte en que el sistema presenta datos históricos o en tiempo real vía web está desarrollada en este lenguaje de programación. Cabe resaltar que junto a PHP se hace uso de otras tecnologías web como HTML, CSS y los applets de Java, sin embargo el rol que cumple la tecnología PHP es de vital importancia, pues nos permite establecer la conexión con la base de datos del sistema y mostrar los datos.

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

PHP es un acrónimo recursivo que significa Pre-Procesador de Hipertexto. Este lenguaje puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

PHP al igual que Java tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows, y puede interactuar con los servidores de web más populares como el anteriormente mencionado Apache.

PHP es una alternativa a las tecnologías no libres de Microsoft ASP y ASP.NET (que utiliza C# y Visual Basic .NET como lenguajes), a ColdFusion de la empresa Adobe, a JSP/Java y a CGI/Perl.

1.9 Modelo Cliente - Servidor:

El sistema “Soft-Control” está diseñado en base a la arquitectura Cliente – Servidor, este modelo de software se basa en el servicio que un servidor le da a uno o varios clientes, es decir el servidor atiende a clientes brindándoles información después de una serie de lógicas de acceso a ella.

Tanto la parte del sistema en planta como el sistema web usan este modelo. Siempre habrá un servidor para la aplicación industrial en la espera de los clientes, los cuales serían los dispositivos que se conectan a él. El sistema siempre tendrá el servidor web en la espera de algún cliente web que pida el acceso a información histórica.

1.9.1 Definición:

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, de bases de datos, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

Para el caso de un sistema SCADA, tanto el computador central como los dispositivos de campo pueden actuar de servidor o de cliente dependiendo de la configuración que se les programe. Pero de cualquier manera usan el modelo Cliente – Servidor independientemente de protocolo de comunicación usados (Serial – Ethernet - OPC).

1.9.2 Ventajas del modelo:

- Centralización del control: los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado o mejorado en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.

La mayoría de los servicios de Internet son tipo de cliente-servidor. La acción de visitar un sitio web requiere una arquitectura cliente-servidor, ya que el servidor web sirve las páginas web al navegador (al cliente). Cuando el navegador web del usuario solicita un artículo particular, el servidor recopila toda la información a mostrar en la base de datos, la articula en una página web, y la envía de nuevo al navegador web del cliente. (Alegsa, 2010)

1.10 Servidor Web

1.10.1 Definición:

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.

El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error, este comportamiento se explica mejor en la Figura 2.

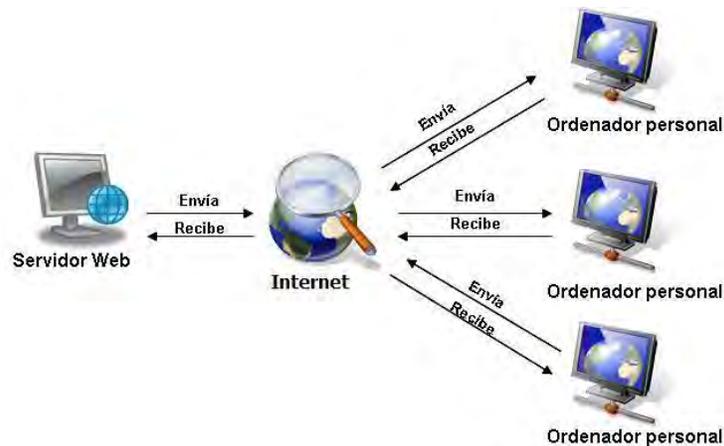


Figura 2 - Modelo Cliente – Servidor (Culturación, 2012)

Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

1.10.2 Aplicaciones en el lado del cliente:

El cliente web es el encargado de ejecutar las aplicaciones, enviadas por el servidor, en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

1.10.3 Aplicaciones en el lado del servidor:

El servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer

ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones. (Breeding Designers, 2009)

El servidor Apache es el más popular entre las computadoras que usan un servidor web.

XAMPP es un paquete de software que contiene distintos servidores agrupados, es independiente de plataforma y además es de software libre. Consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl.



Figura 3 - Xampp y sus aplicativos (Culturación, 2012)

1.11 Base de Datos

Definición:

Una base de datos suele definirse como un conjunto de información organizada sistemáticamente. En la terminología propia de las bases de datos hay tres conceptos claves dentro de las tablas: campo, registro y dato.

MySQL:

MySQL es un sistema de administración de bases de datos (*Database Management System*, DBMS) para bases de datos relacionales. Es la base de datos de código libre más popular y, posiblemente la mejor del mundo. Su continuo desarrollo y su creciente popularidad están haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle.

Lenguaje de Consultas Estructurado

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés *Structured Query Language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de

una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella. (Pérez, 2007)

1.12 Comunicaciones

Los protocolos de comunicación que se implementaron en el sistema SCADA, son: Ethernet, Serial y OPC.

1.12.1 Protocolo Ethernet

Es un protocolo estándar de redes de computadoras de área local. Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.

La Ethernet se tomó como base para la redacción del estándar internacional IEEE 802.3. Usualmente se toman Ethernet e IEEE 802.3 como sinónimos. Ambas se diferencian en uno de los campos de la trama de datos. Las tramas Ethernet e IEEE 802.3 pueden coexistir en la misma red.

Los elementos de una red Ethernet son: tarjeta de red (Figura 4), repetidora, concentradora, puentes, los conmutadores, los nodos de red y el medio de interconexión (

). Para configurar esta comunicación solo se necesita establecer el puerto del servidor y la dirección IP del servidor (Figura 6).



Figura 4 - Tarjeta de red (Academic, 2013)



Figura 5 - Cable Ethernet RJ-45 (Blogspot, 2012)

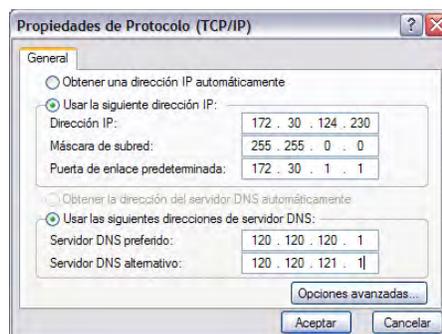


Figura 6 - Configuración del protocolo Ethernet

1.12.2 Protocolo Serial:

Un puerto serie o puerto serial (

) es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente. La comparación entre la transmisión en serie y en paralelo se puede explicar usando una analogía con las carreteras. Una carretera tradicional de un sólo carril por sentido sería como la transmisión en serie y una autovía con varios carriles por sentido sería la transmisión en paralelo, siendo los vehículos los bits que circulan por el cable.



Figura 7 - Puerto serial de una PC (Academic, 2012)

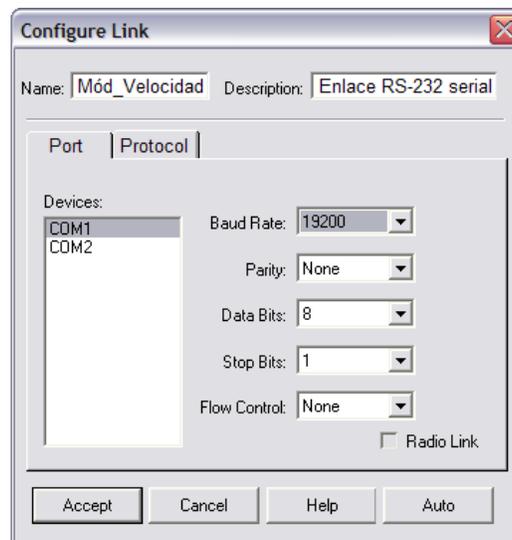


Figura 8 - Configuración de puerto serial

Para poder configurar esta comunicación, se necesita configurar los parámetros (Figura 8): Baudios y el nombre del puerto a usar.

1.12.3 Comunicación OPC:

Actualmente los sistemas SCADA disponen de un tipo de comunicación que se ha convertido en un estándar a nivel internacional para transferir datos independientemente de la aplicación y del lenguaje de comunicación. Dicho estándar es el denominado OPC.

El OPC (*Object Linking and Embedding for Process Control*) es un estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar. De este modo se elimina la necesidad de que todos los programas cuenten con drivers para dialogar con múltiples fuentes de datos, basta que tengan un driver OPC.

Tiene como propósito cubrir las necesidades de acceso en forma estándar de las distintas aplicaciones hacia los dispositivos o base de datos. Es decir una aplicación X y una Y se podrían comunicar con distintos servidores A, B, C de diferentes protocolos de comunicación, siempre y cuando estos tengan interfaces OPC las cuales se les puede aprovechar para conectarlas con las aplicaciones (Figura 9).

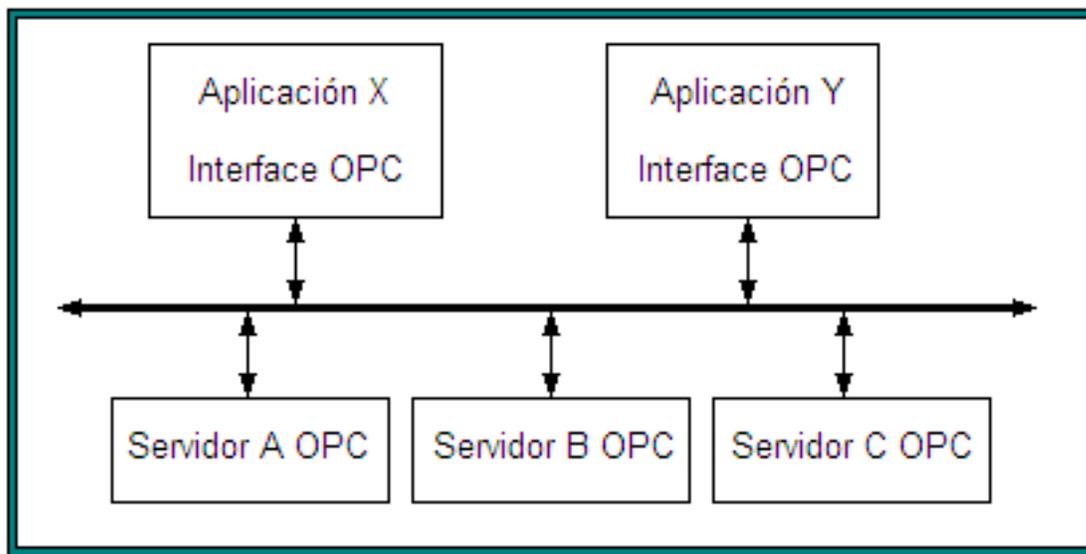


Figura 9 - Funcionamiento comunicación OPC (Udep, 2008)

En la actualidad la mayoría de dispositivos controladores contienen drivers OPC, por tanto no es necesario adaptar los drivers ante nuevos dispositivos de otras marcas.

La arquitectura OPC es de entorno heterogéneo, es decir integra equipos de distintos fabricantes y simplifica las comunicaciones.

Un cliente OPC se puede conectar a múltiples servidores OPC, tan solo direccionándolos. El siguiente ejemplo establece el sistema SCADA como cliente OPC y a los dispositivos como los Servidores OPC, ambos se pueden conectar aun teniendo diferentes protocolos de comunicación gracias a la comunicación OPC (Figura 10).

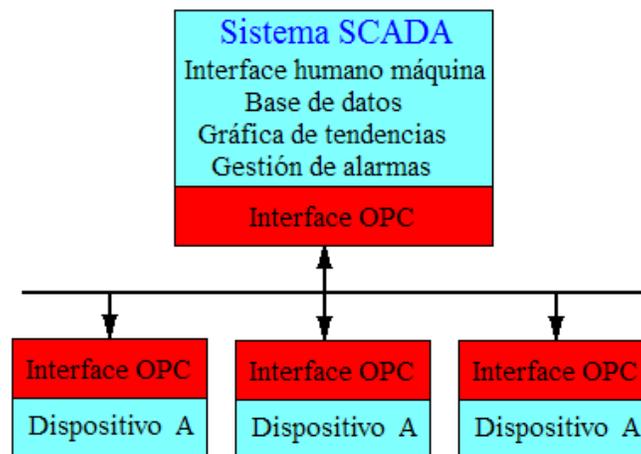


Figura 10 - Comunicación OPC en SCADA (Udep, 2008)

1.12.3.1 Elementos de la comunicación OPC:

El servidor

- Mantiene información sobre el servidor.
- Sirve como contenedor para objetos del grupo OPC.

El grupo

- Mantiene información sobre sí mismo.
- Provee mecanismos para contener/organizar lógicamente ítems.

El elemento

- Representan conexiones a fuentes de datos dentro de un servidor. (Udep, 2008)

Capítulo 2: Diseño

A finales de los años 60, cuando se daba la crisis del software, el hardware dejó de ser un impedimento para el desarrollo de la informática. Había problemas de imprecisión en la planificación, estimación de costos, baja calidad y dificultad de mantenimiento de los programas. Hoy en día eso se dejó atrás gracias a la ingeniería de software la cual es una rama de ingeniería que se enfoca en mejorar la calidad del software en todas sus etapas. Ahora veremos el proceso de diseño del sistema SCADA según la metodología de la ingeniería de software.

2.1 Alcance del software

El software se encarga de monitorear los procesos que se ejecuten en la interfaz de manera remota es decir desde una ubicación dentro de la red de la empresa o desde el acceso en un navegador web. También permitirá controlarlos y modificar sus parámetros, es capaz de almacenar datos de todos los procesos y además presenta reportes históricos de los datos almacenados, el sistema presenta 2 tipos de comunicaciones para trabajar con los dispositivos, la comunicación serial y la comunicación Ethernet, cabe resaltar que el sistema es multiplataforma debido al uso del lenguaje de programación java, por lo tanto el sistema operativo en el cual funcionaría el software no sería una limitación.

En resumen, las funcionalidades son las siguientes:

- Monitoreo desde el sistema y desde un navegador web
- Control de procesos y variación de parámetros
- Adquisición de datos en tiempo real
- Presentación de reportes históricos

2.2 Perspectiva del producto

- Al ser un sistema informático desarrollado para el uso de pymes, se estima que el software tendrá acogida para el control y monitoreo de procesos en empresas en desarrollo que no se requiera de mucha complejidad en el manejo de información.
- Funciona independientemente de otro sistema que posea el departamento.
- Usa tecnologías confiables y con un nivel alto en control de errores, tanto para el software como para la base de datos.
- Dispone de una interfaz gráfica simple para la fácil adaptación para el usuario y a la vez amigable con gráficos representativos en casi todos los paneles.
- El costo del producto sería relativamente bajo respecto a sus competidores en el mercado, pues el sistema fue desarrollado con software libre, con lo cual ayuda que sus costos sean mucho más bajos que los sistemas desarrollados en entornos pagados.

2.3 Capacidades Generales

El sistema será capaz de:

- a) Supervisar los procesos de las diferentes partes de la empresa mediante indicadores y gráficos en tiempo real.
- b) Controlar adecuadamente los mismos y además variar los parámetros de los dispositivos a los cuales estará conectado.
- c) Conectar dispositivos a través de comunicaciones Ethernet y Serial.
- d) Presentar reportes históricos de los datos previamente almacenados
- e) Almacenará datos en una base de datos de MySQL, permitiendo al usuario especificar el tiempo de guardado.

2.4 Restricciones generales

- El sistema trabaja independientemente de otros sistemas, es decir no está prevista la integración con otro tipo de software.
- Existe un protocolo diseñado para la comunicación entre el sistema y los dispositivos a conectar.
- El sistema se ejecuta con la base de datos y la aplicación en una sola capa, sin embargo el acceso por web se puede hacer desde cualquier punto en que se encuentre el navegador.
- Los operadores de las variables son los básicos, es decir se puede emplear funciones que usen suma, resta, multiplicación, división, exponentes y radicales.

- Se podrá manejar hasta 6 variables por proceso independiente del tipo (Entrada o Salida).

2.5 Entorno operacional

El sistema funciona en cualquier sistema operativo, gracias al lenguaje java, además es una aplicación *Stand-Alone*, es decir de escritorio, es necesario que el paquete de software Microsoft Office sea instalado en el equipo en que funcionará el sistema, el cual usa el aplicativo Microsoft Excel como medio para exportar reportes, además usa un servidor web llamado “Apache” el cual es también un software libre, como el acceso al sistema no tendrá mucha concurrencia no es necesario contar con un dominio, se puede usar solo una dirección “IP” pública y guardarla en el navegador, para el acceso al sistema se le brindará al o los usuarios una cuenta de acceso, al igual que otra cuenta diferente para el acceso web.

2.6 Características de los Usuarios

El usuario del sistema será un operario del área de control o un supervisor, dicho personal necesita conocer los valores de los parámetros del proceso que se le encarga para así interactuar con el sistema y lograr un control óptimo. El usuario además debe disponer conocimiento de los tipos de comunicación usados en el proceso y de conocimientos básicos de computación básica.

2.7 Modelo de Casos de Uso

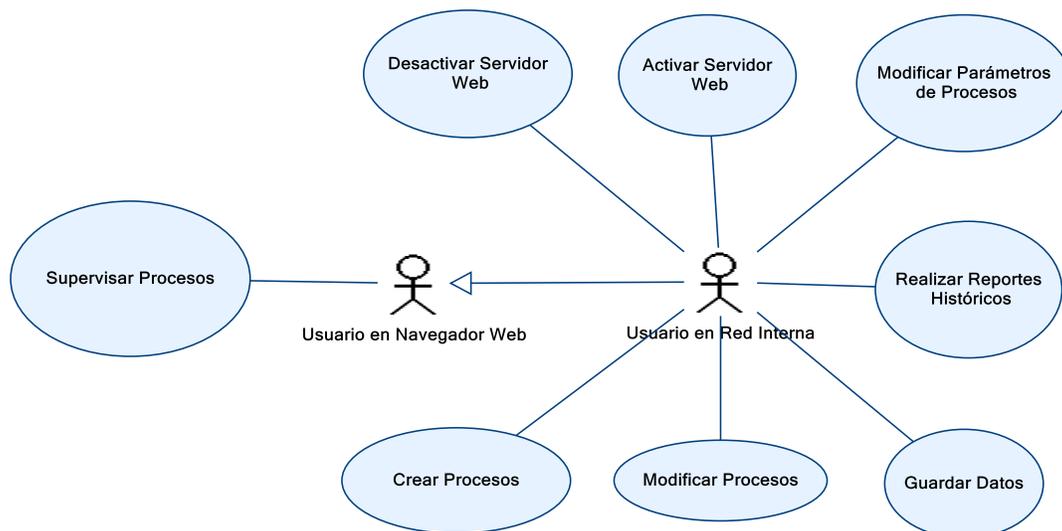


Figura 1 - diagrama de casos de USO

2.8 Especificación Textual

Nombre	Supervisa Proceso
Actores	Usuario interno y usuario web.
Objetivo	Supervisa mediante un histograma el estado del proceso en tiempo real.
Precondiciones	El usuario debe haber iniciado sesión en el sistema y abierto el proceso a supervisar.
Postcondiciones	Proceso bajo supervisión
Escenario básico	<ul style="list-style-type: none"> • Iniciar sesión como usuario interno • Seleccionar proceso • Conectar dispositivos • Abrir histograma

Nombre	Activar Servidor Web
Actores	Usuario interno
Objetivo	Activar la supervisión de procesos mediante el navegador web.
Precondiciones	Se debe tener activado por lo menos un proceso antes de activar el acceso mediante web.
Postcondiciones	El acceso mediante el navegador web se habilita y ahora es posible visualizar en tiempo real los procesos por la web.
Escenario básico	<ul style="list-style-type: none"> • El usuario interno inicia sesión • Se seleccionan los procesos • Se conectan los dispositivos • Se activa el servidor web

Nombre	Desactivar Servidor Web
Actores	Usuario interno
Objetivo	Desactivar la supervisión de procesos mediante el navegador web.
Precondiciones	Se debe tener activado el acceso mediante web.
Postcondiciones	El acceso mediante el navegador web se deshabilita y ahora no es posible visualizar en tiempo real los procesos por la web.
Escenario básico	<ul style="list-style-type: none"> • Se desactiva el servidor web

Nombre	Modificar Parámetro de Proceso
Actores	Usuario interno
Objetivo	Controlar el proceso mediante sus parámetros.
Precondiciones	Proceso previamente abierto e iniciado comunicaciones con los dispositivos
Postcondiciones	Proceso cambiante al estado determinado por sus respectivos parámetros.
Escenario básico	<ul style="list-style-type: none"> • Usuario abre el proceso • Ingresa valores en variables de salida • Envía los valores a los dispositivos • El proceso es modifica en función a los nuevos parámetros

Nombre	Realizar Reporte Histórico
Actores	Usuario interno
Objetivo	Estudiar el comportamiento del proceso en un determinado periodo
Precondiciones	El usuario ha iniciado sesión en el sistema Los datos del proceso ya han sido guardados previamente
Postcondiciones	El reporte es generado
Escenario básico	<ul style="list-style-type: none"> • Usuario inicia sesión en el sistema • Selecciona la opción reporte • Selecciona la fecha de inicio y de fin • Selecciona el proceso y la o las variables • Se genera el reporte

Nombre	Guardar Datos
Actores	Usuario interno
Objetivo	Almacenar de manera periódica los datos de los procesos
Precondiciones	El usuario debe haber iniciado sesión el sistema. Debe haber seleccionado el proceso Debe indicar el tiempo de guardado
Postcondiciones	El proceso almacena los datos según la frecuencia indicada
Escenario básico	Usuario inicia sesión Usuario abre el proceso Puede elegir conectar antes o después Selecciona la opción almacenar datos

Nombre	Modificar Proceso
Actores	Usuario interno
Objetivo	Modificar datos de proceso como el nombre, descripción y permite además modificar los datos de sus variables como nombre, alarmas, guardado y graficado.
Precondiciones	El proceso ha sido previamente guardado Se sugiere que el proceso haya sido detenido previamente
Postcondiciones	Proceso modificado a las condiciones definidas por el usuario
Escenario básico	Usuario ingresa al sistema Se selecciona el proceso Se selecciona la ventana de variables Se modifica los valores requeridos Se selecciona la opción de guardar

Nombre	Crear Proceso
Actores	Usuario interno
Objetivo	Añadir un proceso al sistema
Precondiciones	Usuario inicia sesión al sistema Conectar correctamente los dispositivos
Postcondiciones	El proceso se crea satisfactoriamente y es posible ponerlo en marcha
Escenario básico	Usuario inicia sesión Selecciona la opción crear proceso Ingresa el nombre, el tipo de comunicación, sus parámetros y variables del proceso Selecciona la opción aceptar

2.9 Modelo Conceptual

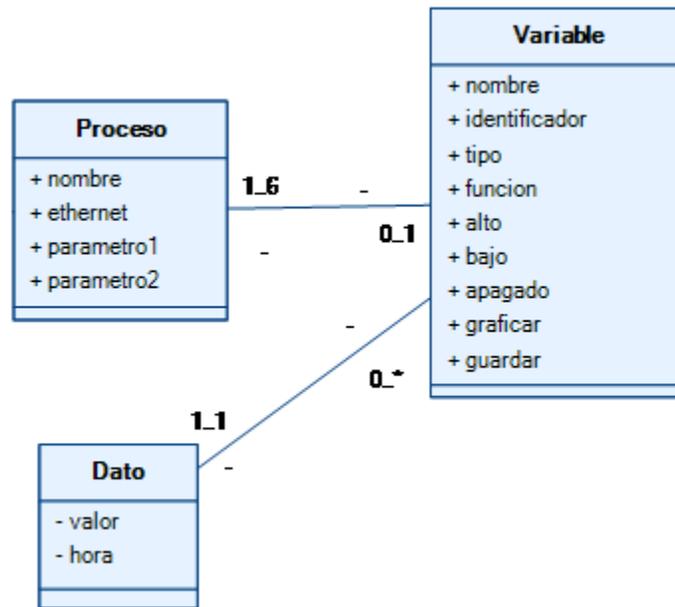


Figura 2 - Modelo Conceptual

El sistema maneja el concepto de proceso como un grupo de variables comunicadas con dispositivos externos, es por eso que el sistema necesita almacenar aquellos datos que son imprescindibles para ellos como el nombre, el tipo de comunicación (Ethernet o serial) y los dos parámetros los cuales serían el puerto y los baudios para la comunicación serial y la dirección IP y el puerto para la comunicación Ethernet, por otro lado el proceso posee 6 variables como máximo, cada una de ellas posee su nombre, su identificador (para el protocolo de comunicación), el tipo de variable que indica si es tipo Input(Entrada) u Output(Salida), también posee una función aritmética que le facilita al programador del proceso a modelar los procesos, de la misma manera posee valores para las alarmas que indiquen un alto, bajo y un nivel apagado. Por último posee un indicador de almacenamiento y de gráfica en tiempo real.

Cada variable de los procesos posee datos, los cuales se almacenan en la base de datos con fecha y su valor correspondiente. Además en la base de datos también se almacenan los datos de los usuarios del sistema, es decir las cuentas para los usuarios web y para los que tienen acceso directo desde el programa principal.

2.10 Modelo relacional de la base de datos del sistema

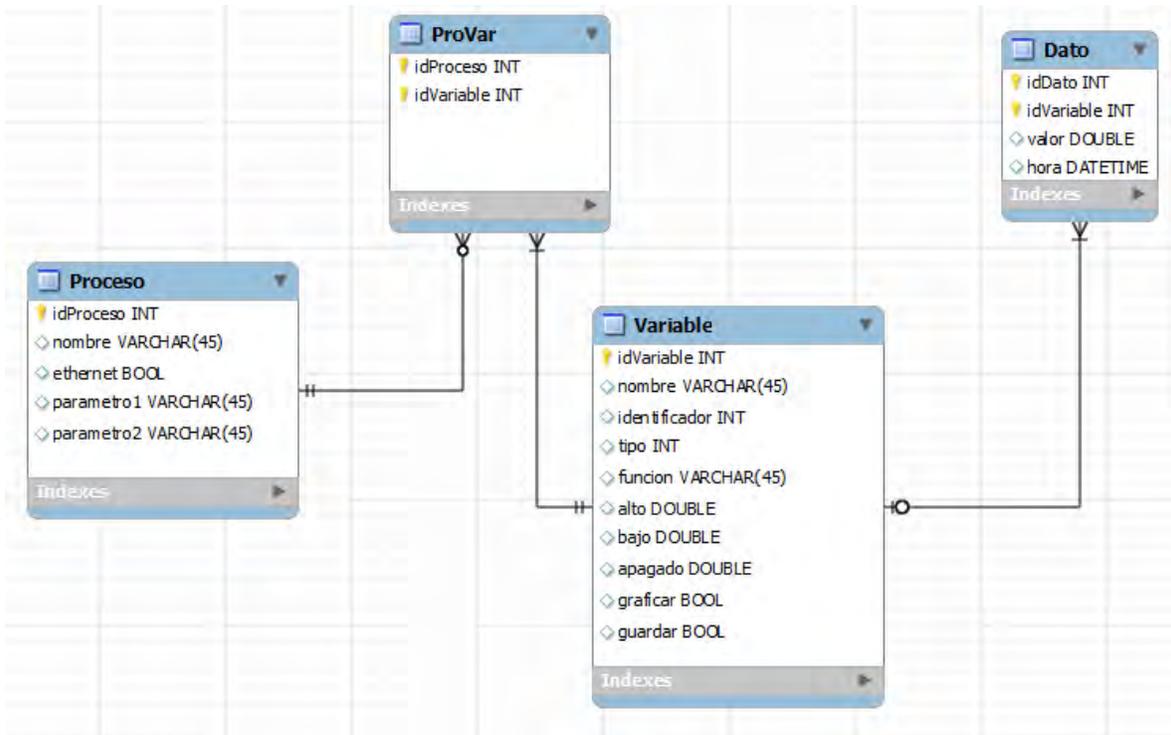


Figura 3 - Modelo Relacional

2.10.1 Diccionario de datos de las tablas

Dato			
Column name	DataType	PK	NN
idDato	INT	✓	✓
idVariable	INT	✓	✓
valor	DOUBLE		
hora	DATETIME		

ProVar			
Column name	DataType	PK	NN
idProceso	INT	✓	✓
idVariable	INT	✓	✓

Proceso			
Column name	DataType	PK	NN
idProceso	INT	✓	✓
nombre	VARCHAR(45)		
ethernet	TINYINT(1)		
parametro1	VARCHAR(45)		
parametro2	VARCHAR(45)		

Variable			
Column name	DataType	PK	NN
idVariable	INT	✓	✓
nombre	VARCHAR(45)		
identificador	INT		
tipo	INT		
funcion	VARCHAR(45)		

alto	DOUBLE		
bajo	DOUBLE		
apagado	DOUBLE		
graficar	TINYINT(1)		
guardar	TINYINT(1)		

2.10.2 Detalle del modelo relacional del Sistema:

Lo que el sistema necesita almacenar son principalmente los datos de los procesos, pero para que dichos datos se almacenen de forma ordenada se necesita guardar los procesos y variables correspondientes, es por eso que se creó una tabla Proceso con los parámetros Nombre, Ethernet (Tipo de comunicación true para Ethernet, false para Serial), el parámetro1 que puede ser el IP o los baudios y el parámetro2 que es el puerto según el tipo de comunicación.

Como el proceso puede tener muchas variables y la misma variable puede repetirse en otro proceso, se creó una tabla intermedia ProVar para relacionar tanto la tabla Proceso como la tabla Variable. La tabla variable almacena los datos de las variables, los cuales son el nombre, identificador, tipo, función, alarmas, almacenamiento y graficado.

Por último la tabla de datos la principal del sistema, posee los campos de valor y fecha del dato almacenado, esa tabla es la que nos permite hacer los reportes históricos.

2.11 Arquitectura Lógica

2.11.1 Especificación de las capas

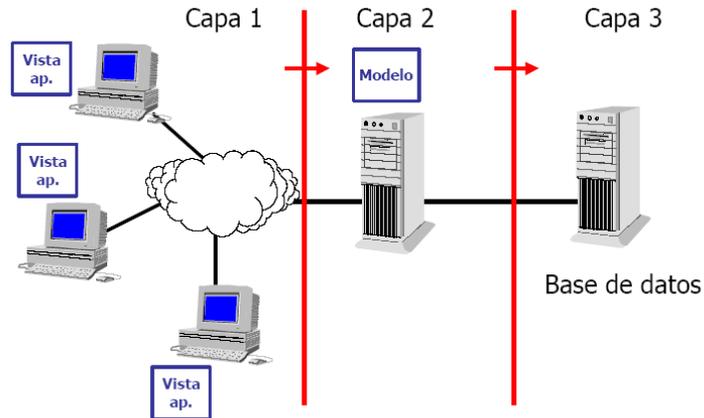


Figura 4 - Arquitectura a 3 capas (Oness, 2012)

El sistema posee una arquitectura de tres capas (Figura 4), como puede observarse en la imagen la primera capa consta del sistema instalado en planta con la lógica de negocios, la segunda capa sería el servidor de la base de datos, por último la interfaz gráfica en el navegador web que monitorea el proceso, éste accede desde internet por lo tanto no necesita encontrarse en un área cercana al sistema.

2.11.2 Diagrama de Componentes

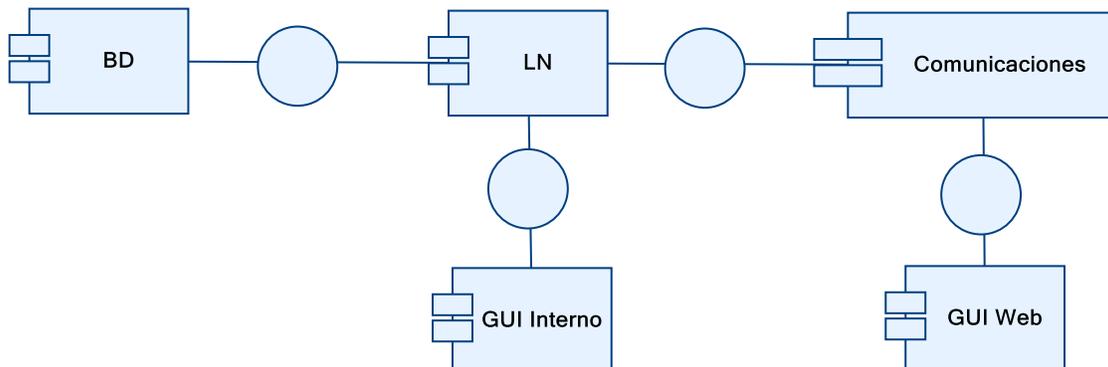


Figura 5 - Diagrama de componentes del sistema

El sistema tendría los componentes de base de datos (BD), lógica de negocio (LN), interfaz gráfica de usuario interna y web (GUI), además de la interfaz de comunicación para los distintos protocolos que manejaría el sistema. Dichos componentes estarían relacionados entre sí.

2.12 Arquitectura Física

En lo que respecta a la arquitectura física la computadora que tendría instalado el sistema debería tener por lo menos las siguientes características:

- Procesador Pentium IV (2.2 GHz)
- 512 Mb RAM
- Disco duro de 80 Gb.
- Máquina virtual de java
- Paquete Microsoft Office
- Mouse óptico
- Teclado

2.12.1 Especificación de las herramientas tecnológicas y herramientas a utilizar

El sistema hace uso de Microsoft Excel (con el fin de exportar reportes del proceso) es una aplicación distribuida por Microsoft office para hojas de cálculo. Este programa es desarrollado y distribuido por Microsoft, y es utilizado normalmente en tareas financieras y contables. El sistema operativo más estable para su uso es Windows XP que es una versión de Microsoft Windows, línea de sistemas operativos desarrollado por Microsoft. El sistema debe tener los siguientes requisitos:

Tabla 1 - Tabla de herramientas tecnológicas

	Mínimos	Recomendados
Procesador	233 MHz	233 MHz o superior
Memoria	64 Mb de RAM	128 Mb de RAM o superior
Vídeo	Super VGA (800 x 600) o superior	
Espacio en disco duro	1,5 GB o superior (se necesitan 1.8 GB más para el Service Pack 2 y otros 900 MB adicionales para el Service Pack 3)	
Dispositivos ópticos	Unidad de CD-ROM o DVD-ROM	
Periféricos	Teclado y mouse u otro dispositivo señalizador	
Multimedia	Tarjeta de sonido, altavoces o auriculares	

El lenguaje de programación que se usó para desarrollar el sistema fue JAVA. El lenguaje para la programación en Java, es un lenguaje orientado a objeto, de una plataforma independiente.

Esta programación Java tiene muchas similitudes con el lenguaje C y C++, así que si se tiene conocimiento de este lenguaje, el aprendizaje de la programación Java será de fácil comprensión por un programador que haya realizado programas en estos lenguajes.

2.12.2 Diagrama de Despliegue

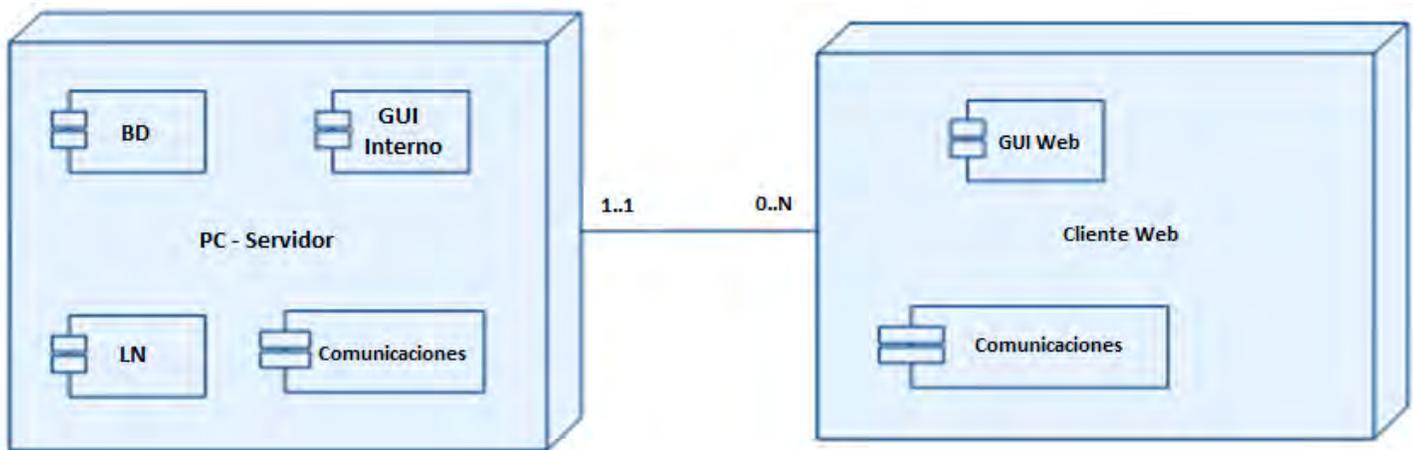


Figura 6 - Diagrama de despliegue del sistema

El sistema trabajará con 2 nodos principalmente (puede trabajar con 3 para mayor seguridad), el primer nodo contendrá el servidor de aplicaciones, web y de base de datos, se ejecutarán las aplicaciones respectivas a la espera de los clientes. El otro nodo estará compuesto tanto de los clientes web o clientes “internos” los cuales vendrían a ser las tarjetas o dispositivos que se conectarán directamente al sistema.

2.13 Diseño de la Interfaz

Una parte importante en el desarrollo del software es el diseño de la interfaz, la cual debe permitir al usuario observar la mayor cantidad de información y brindarle un fácil acceso a esta. La interfaz debe ser “amigable” para que el usuario no sienta resistencia al cambio y se acostumbre al software mucho más rápido.

Acceso al Sistema

El sistema presenta una interfaz de acceso por nombre de usuario y contraseña para el acceso al sistema.

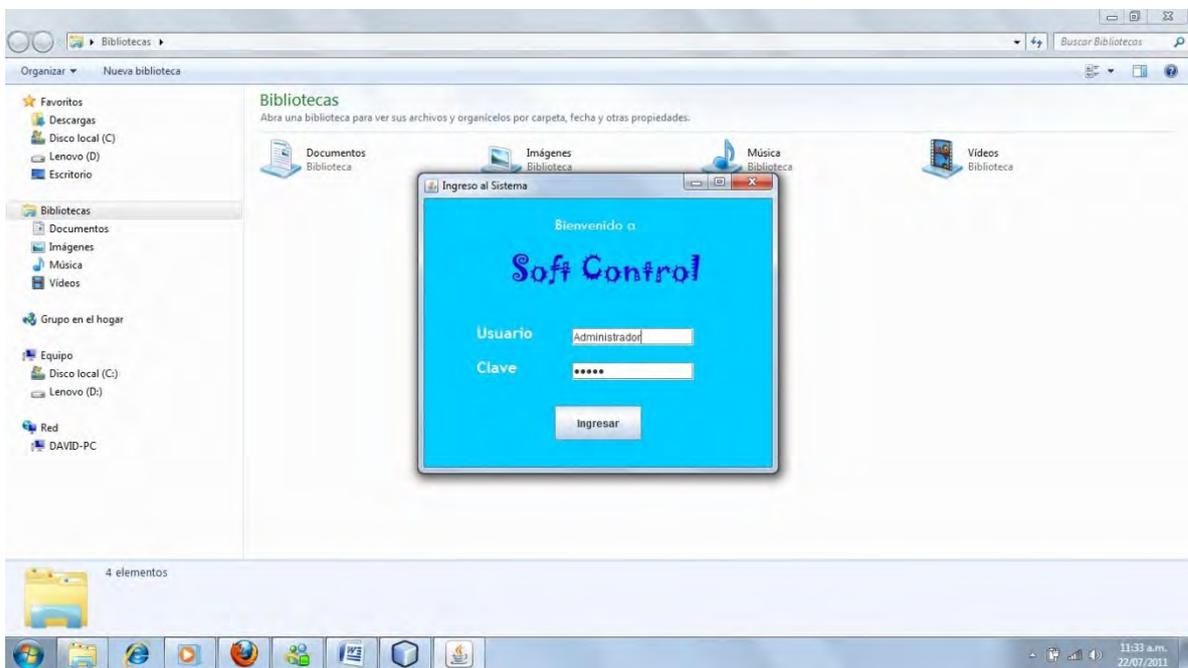


Figura 7 - Módulo de acceso al sistema

Pantalla Principal

La siguiente imagen muestra la pantalla de bienvenida, nótese que se muestra un mensaje de confirmación de la base de datos.

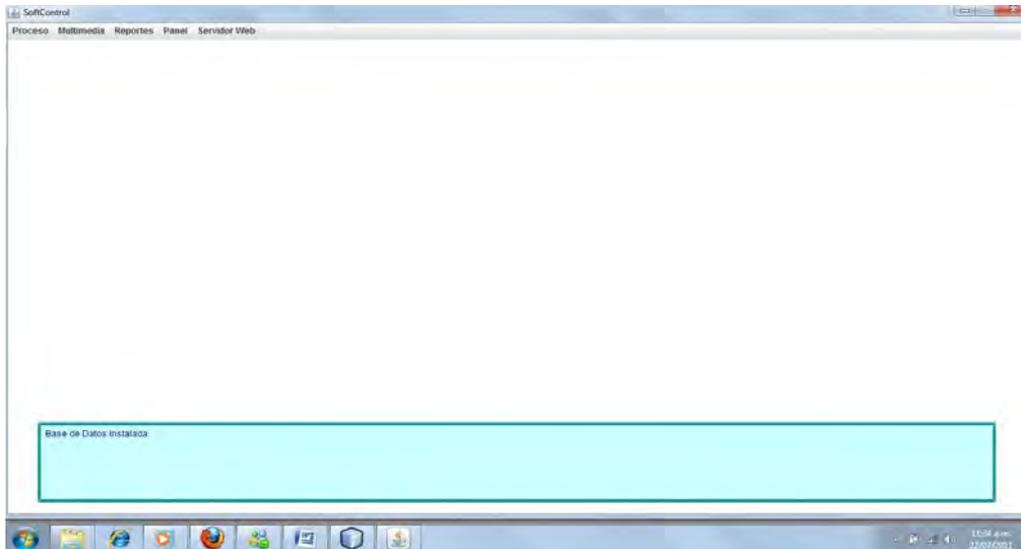


Figura 8 - Pantalla de bienvenida

Menú Principal

En el menú de la pantalla principal se encuentran los siguientes botones, cada uno cumple la función descrita en la especificación del diagrama de clases del diseño (ver más adelante).

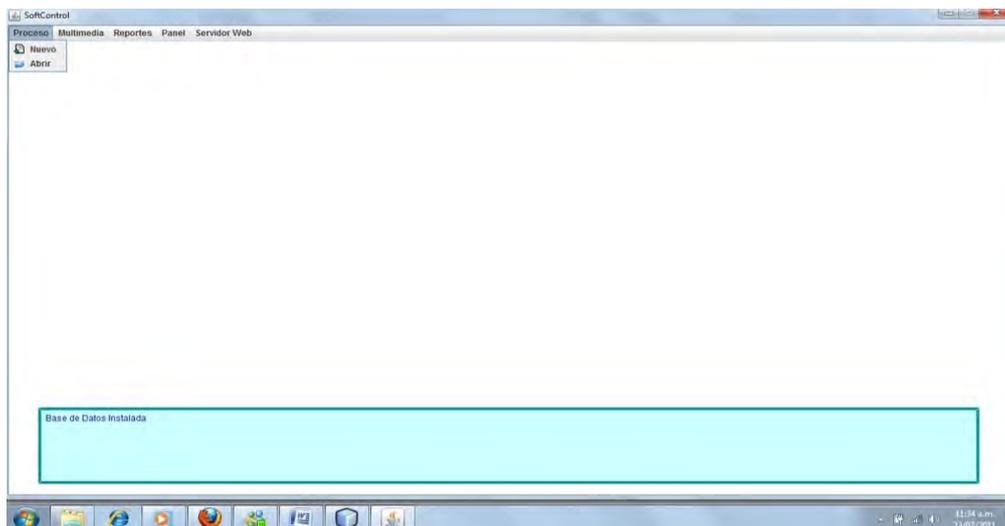


Figura 9 - Menú principal

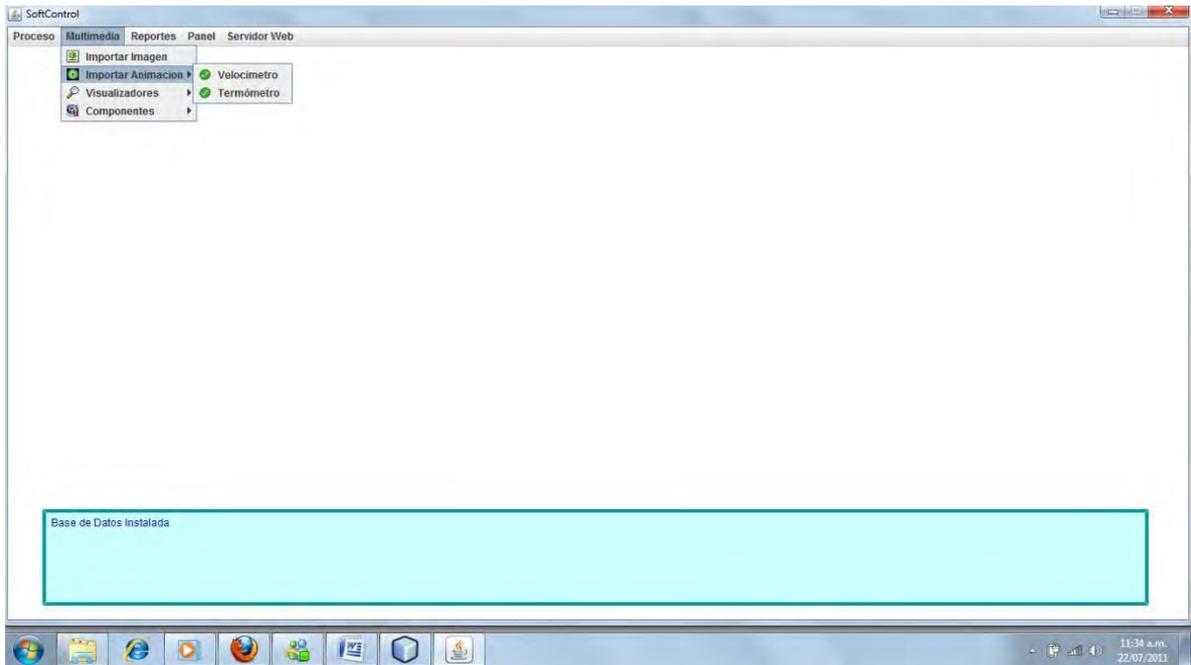


Figura 10 - Pestaña multimedia

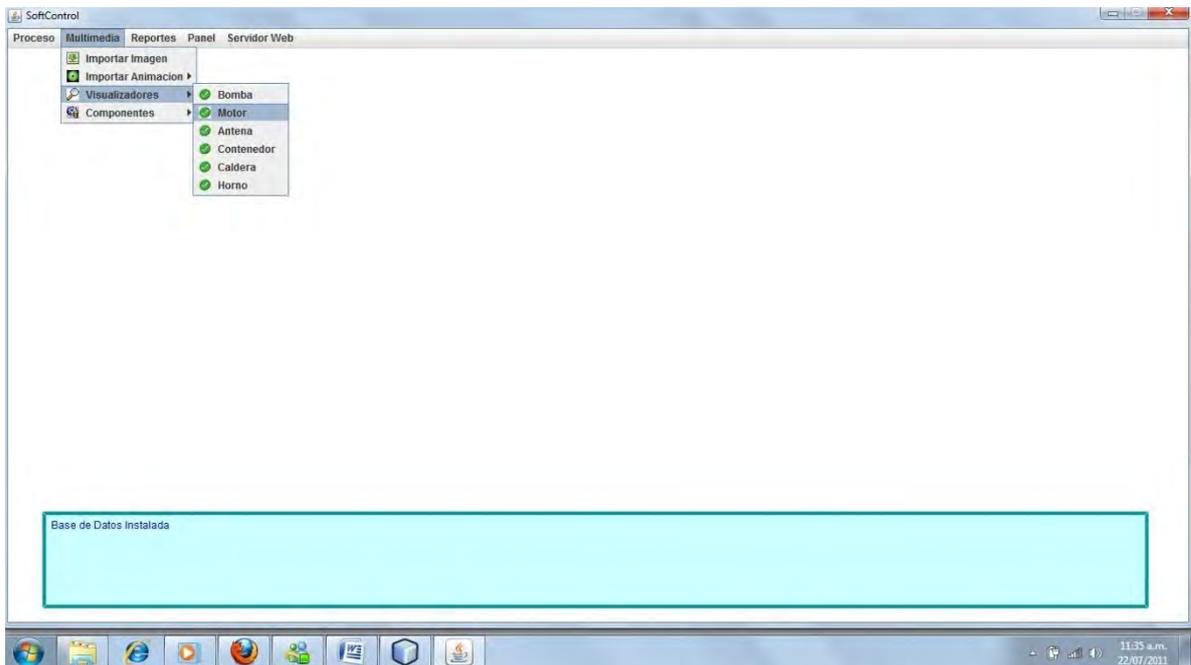


Figura 11 - Visualizadores de variables

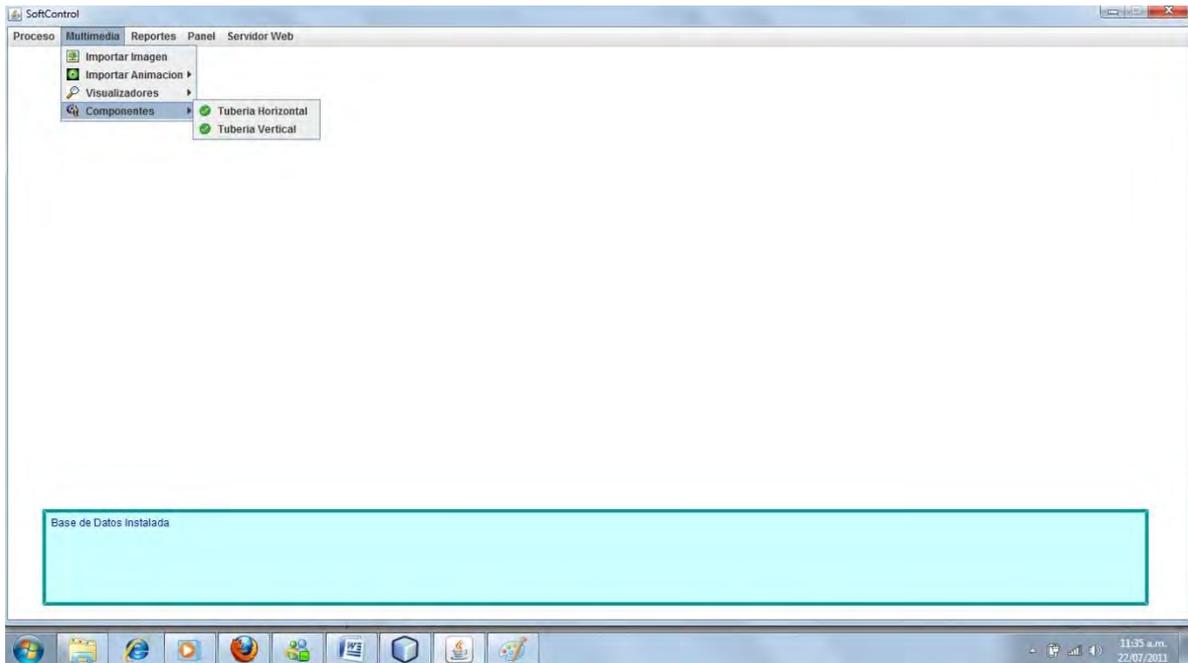


Figura 12 - Componentes de conexión de proceso

Reportes

Desde la ventana principal del sistema se puede acceder a la interfaz para realizar cualquier reporte de datos históricos previamente guardados.

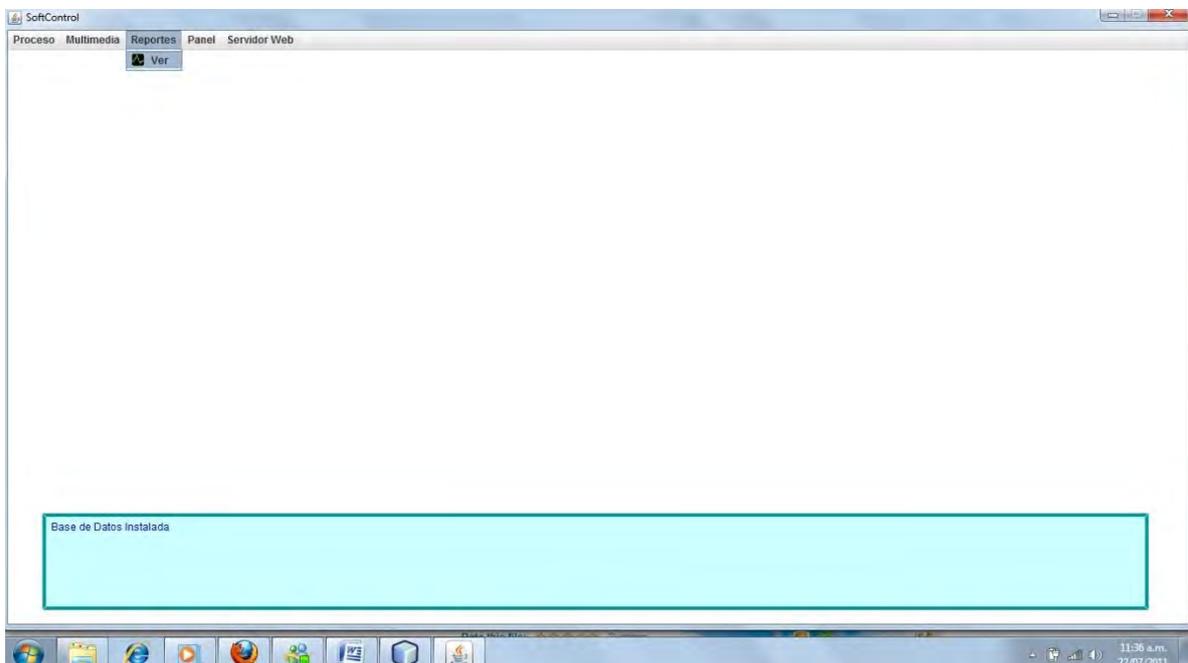


Figura 13 - Opción reportes

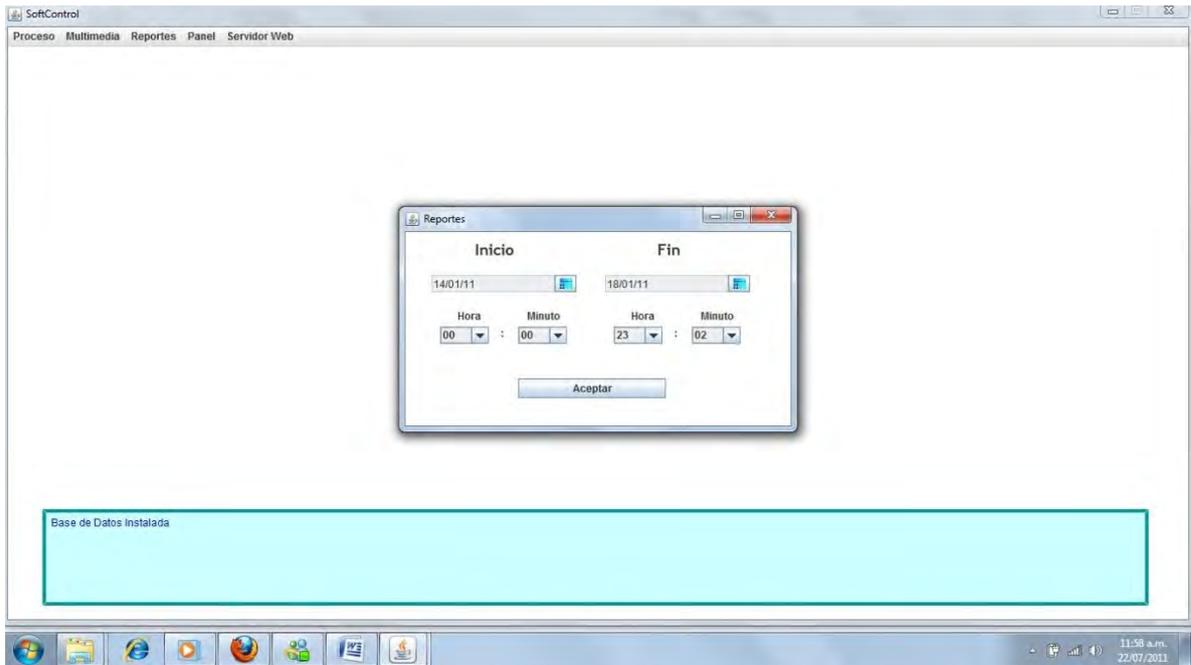


Figura 14 - Selección de fechas para reporte

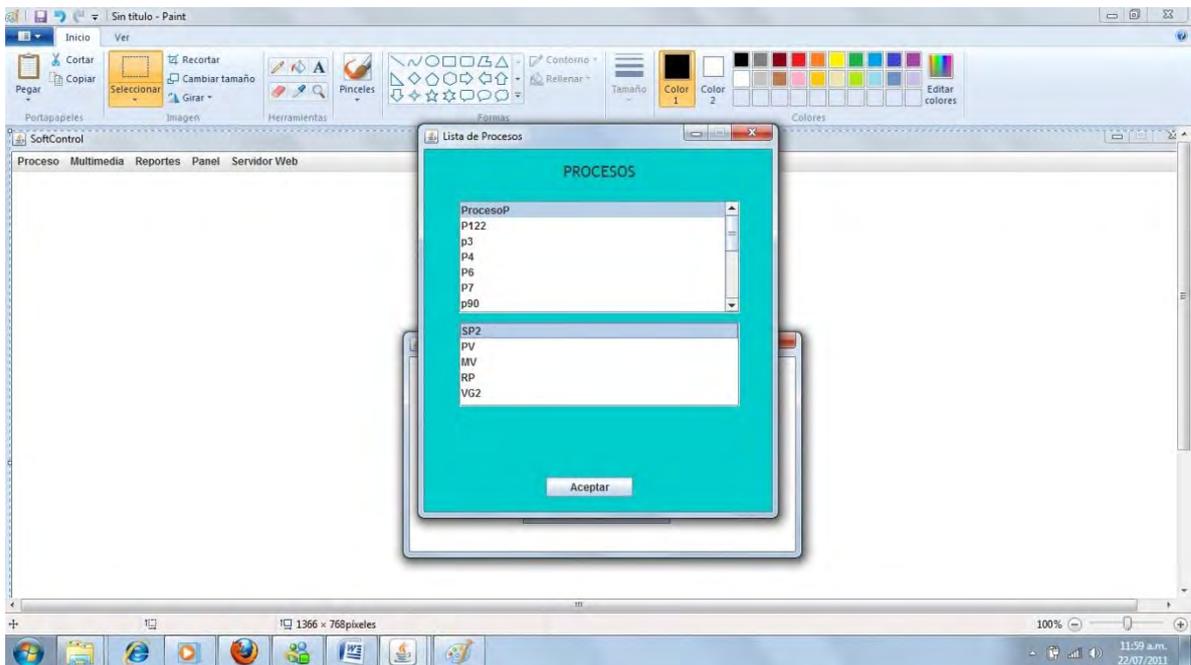


Figura 15 - Selección de proceso a reportar



Figura 16 - Reporte generado

Zoom en reporte



Figura 17 - Selección de área de zoom



Figura 18 - Zoom de área sombreada

Cambiando Color al Panel Principal

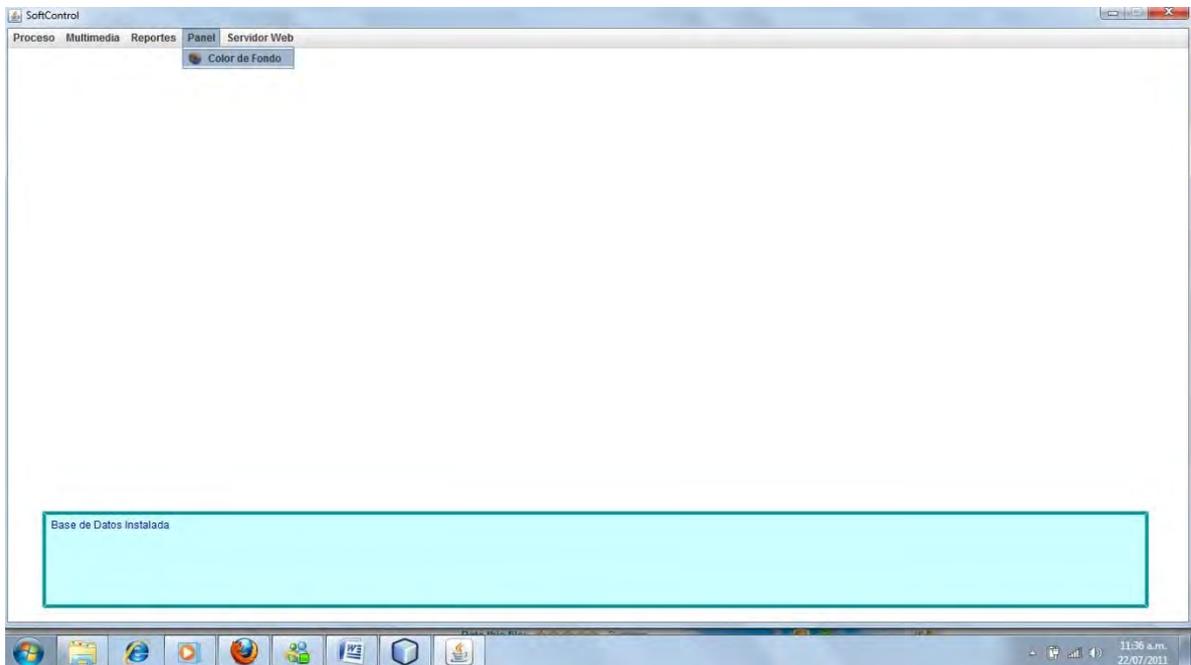


Figura 19 - Opción de cambiar color de fondo

Activando Servidor Web

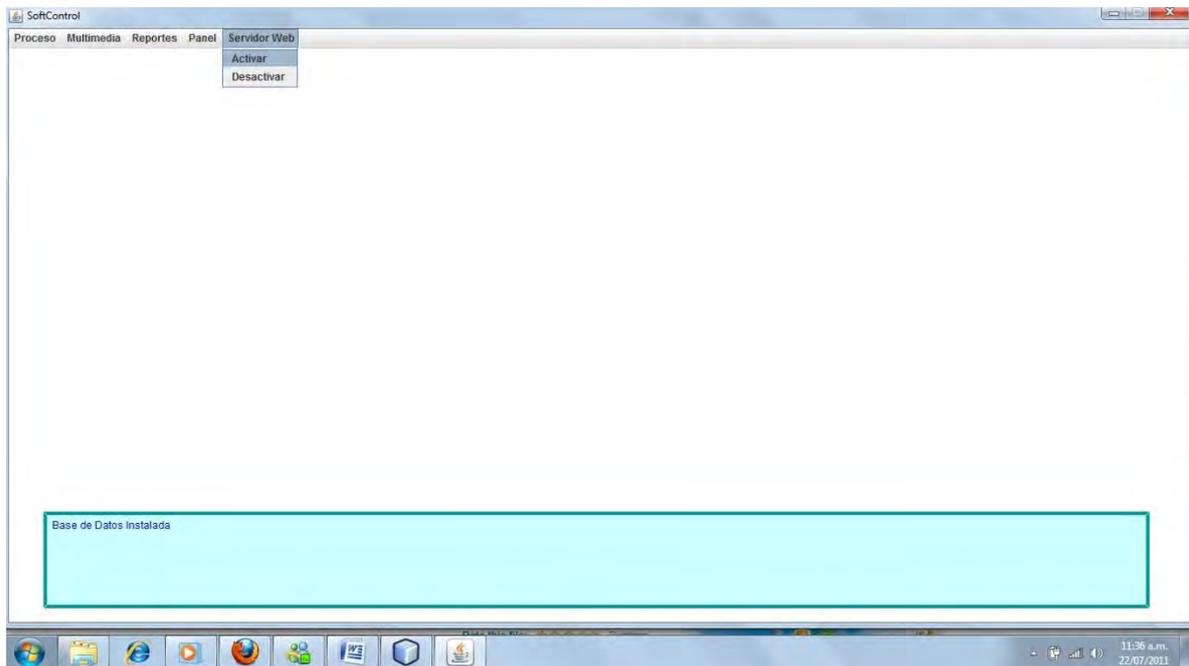


Figura 20 - Opción de comunicación web

Creando nuevo proceso:

En el menú entramos a Proceso → Nuevo y procedemos a crear el proceso

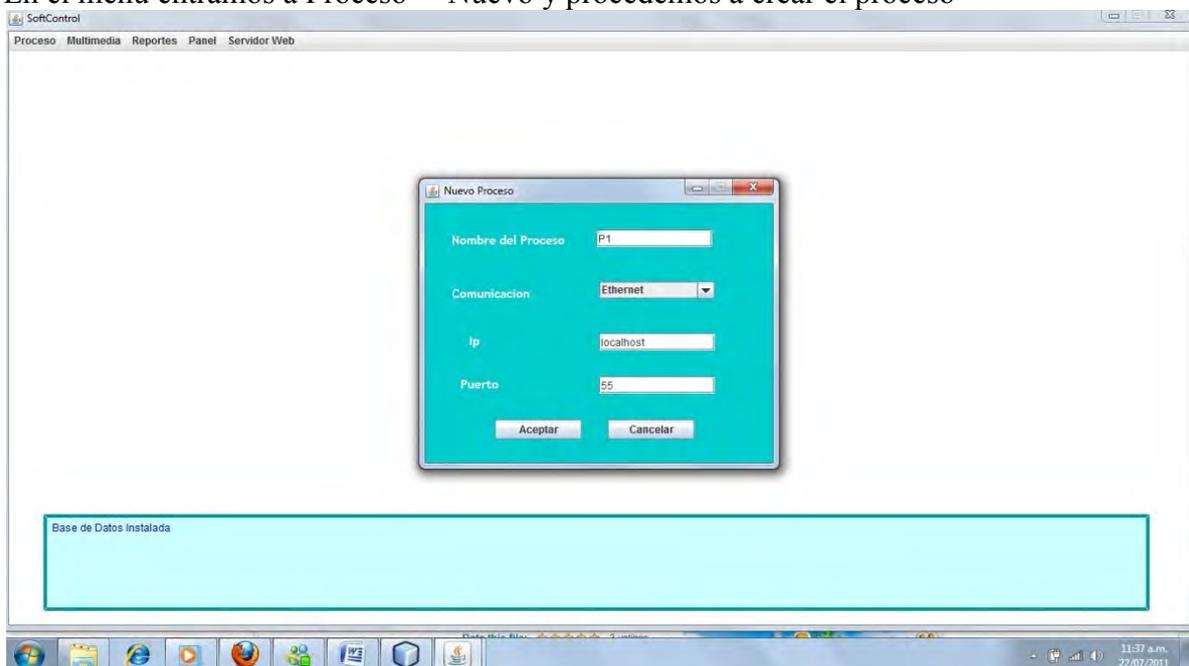


Figura 21 - Ventana de nuevo proceso

Luego se procede a editar las variables del proceso

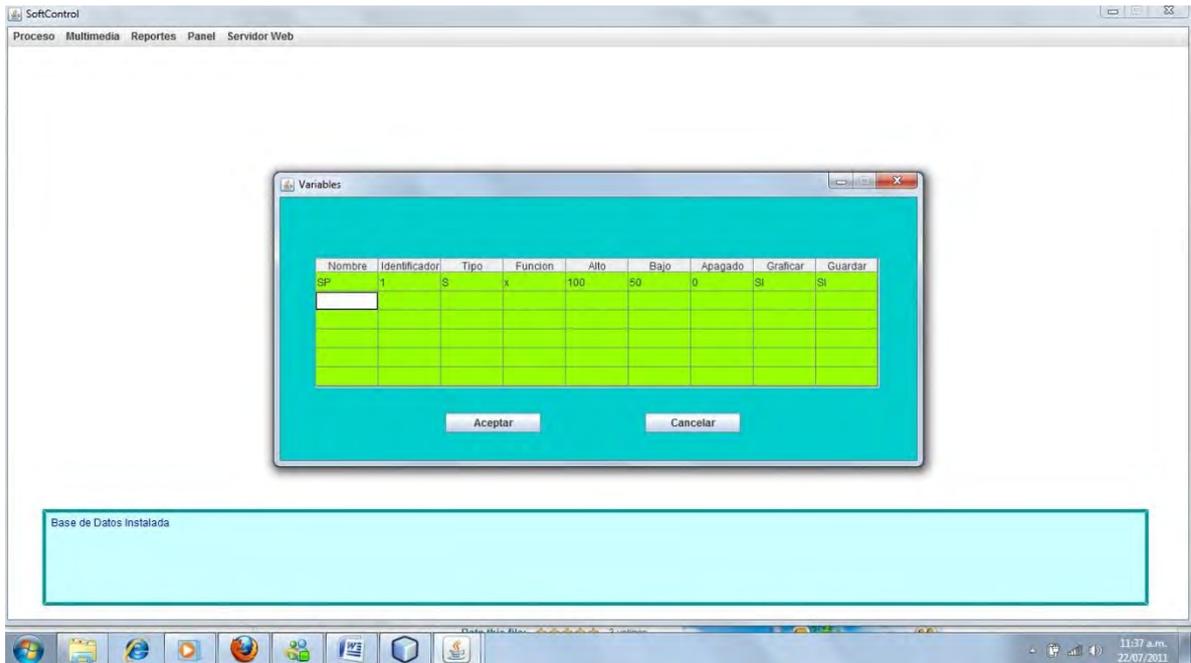


Figura 22 - Configuración de variables

También es posible abrir un proceso previamente guardado en la base de datos:

Entramos en el menú principal, en Proceso → Abrir y seleccionamos el proceso.

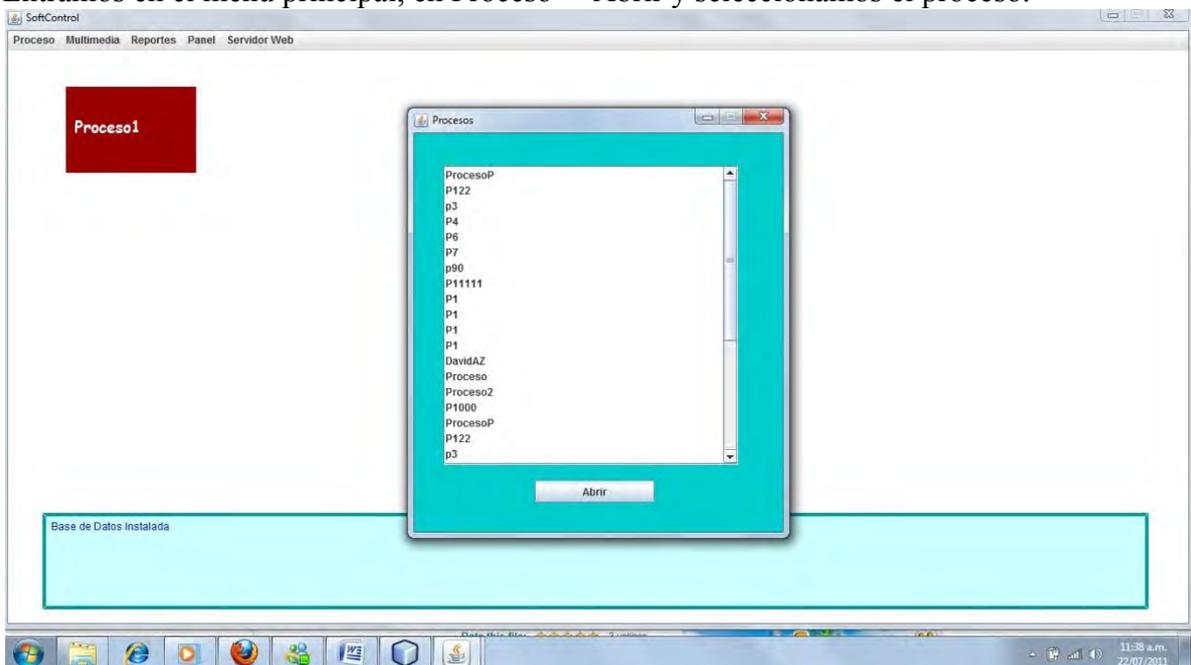


Figura 23 - Ventana para abrir proceso a supervisar

El panel principal del proceso

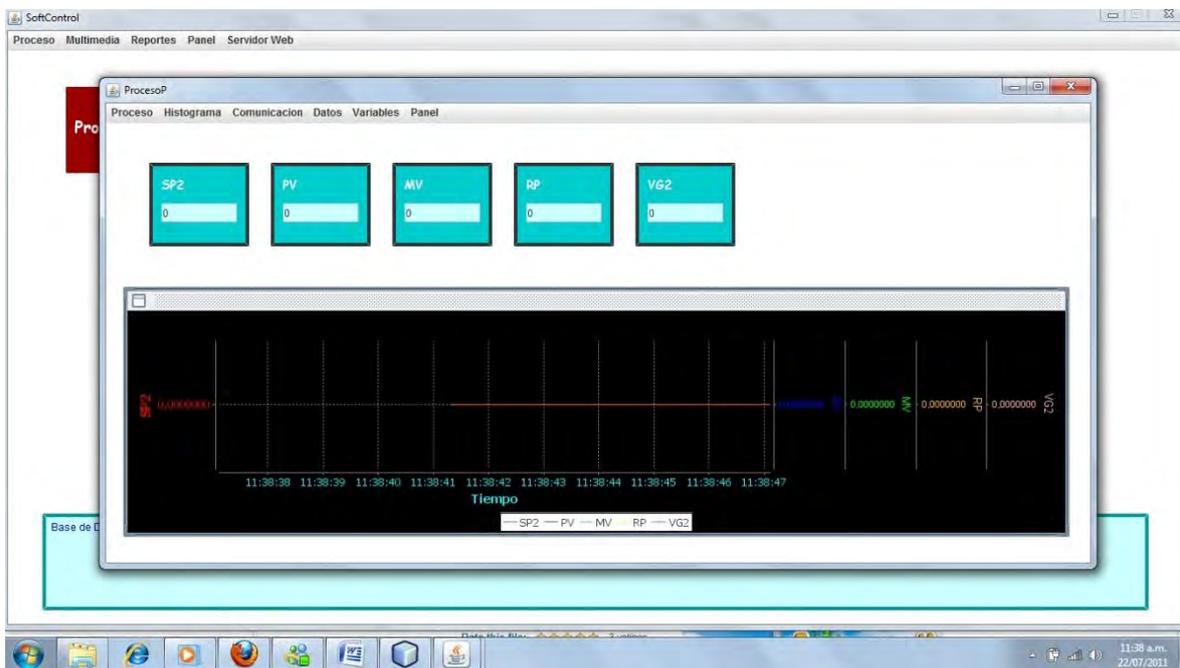


Figura 24 - Panel principal de proceso

El menú del panel de proceso:

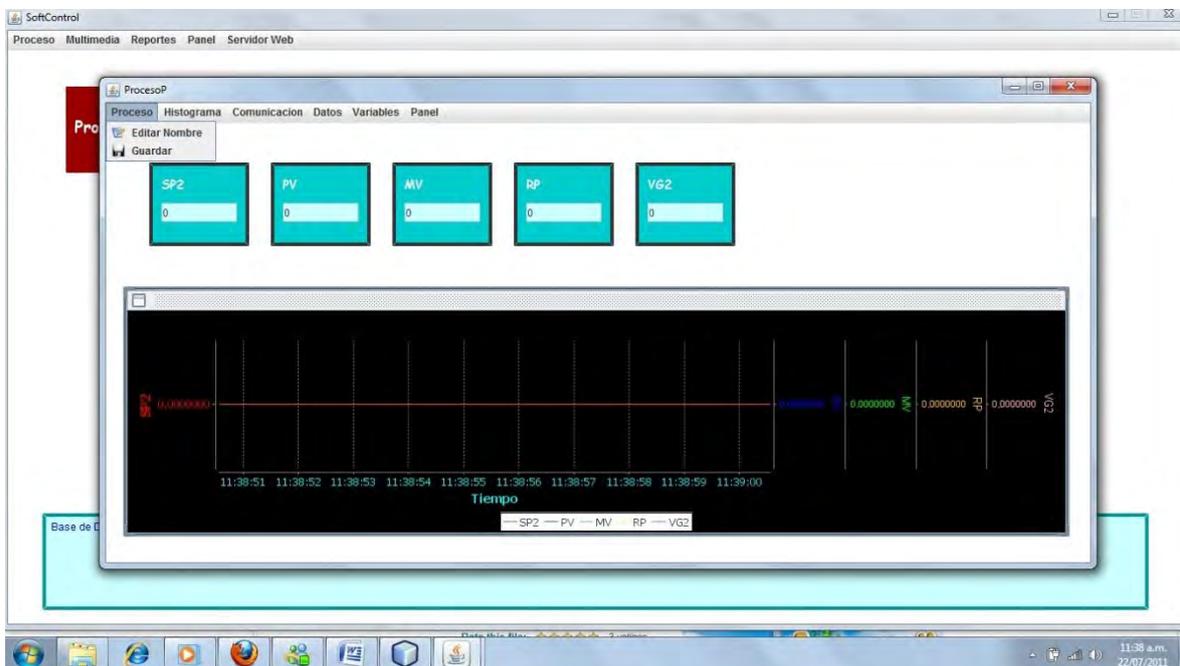


Figura 25 - Menú de proceso

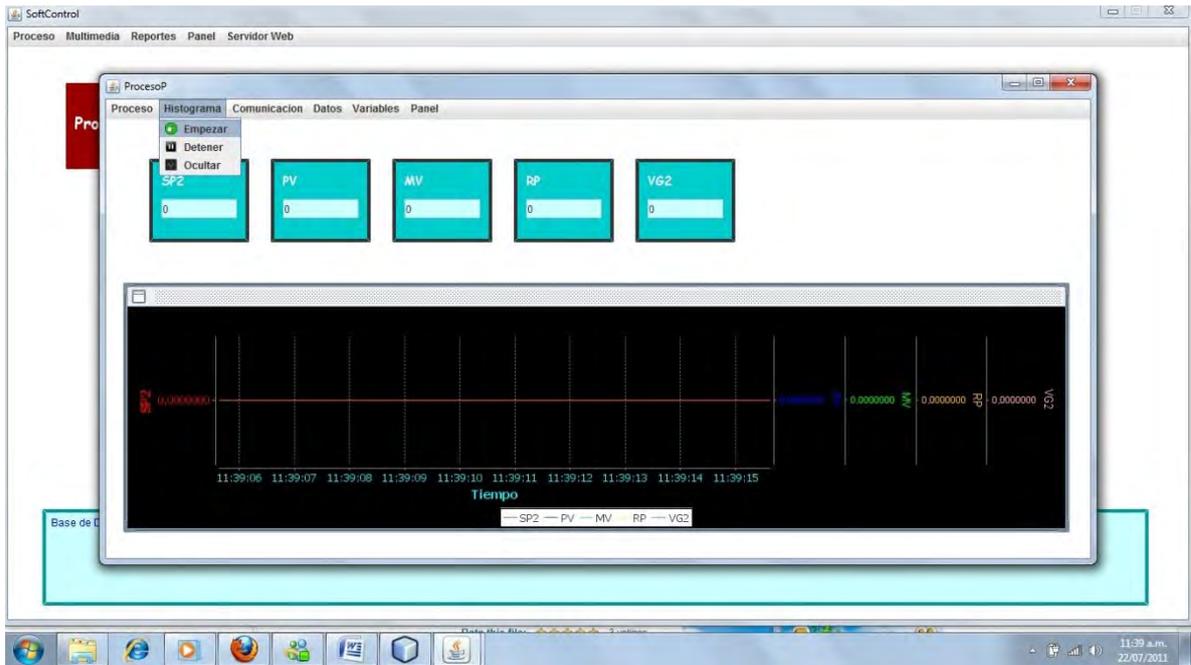


Figura 26 - Histograma en tiempo real

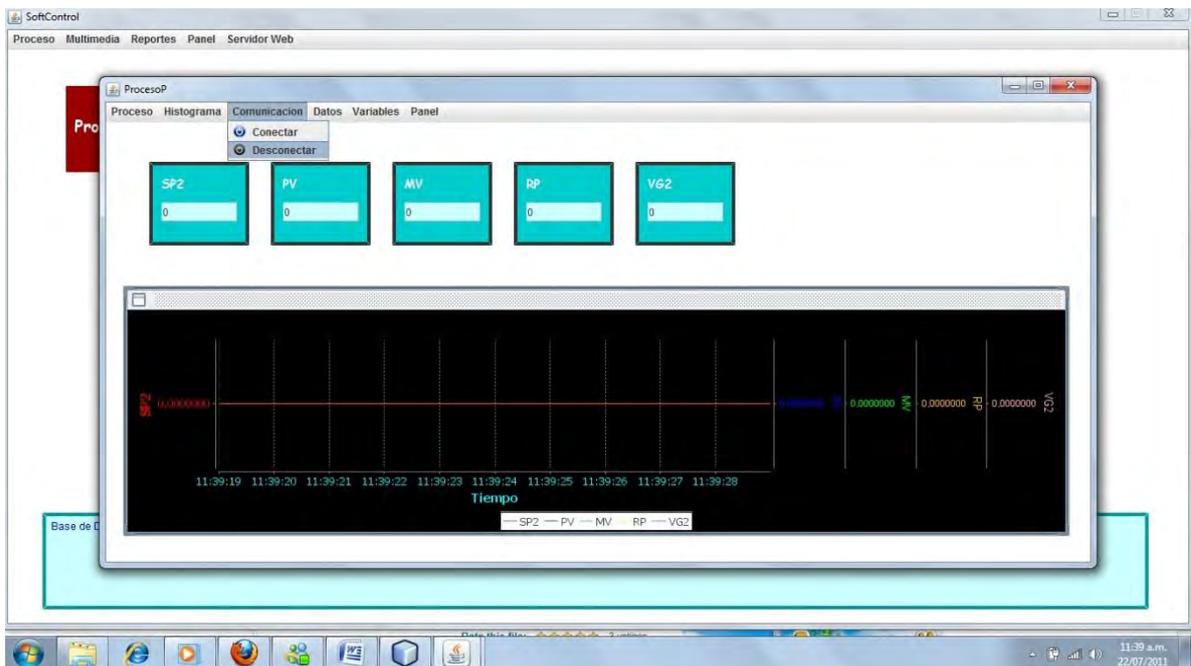


Figura 27 - Comunicación del proceso

Recibiendo Datos del proceso

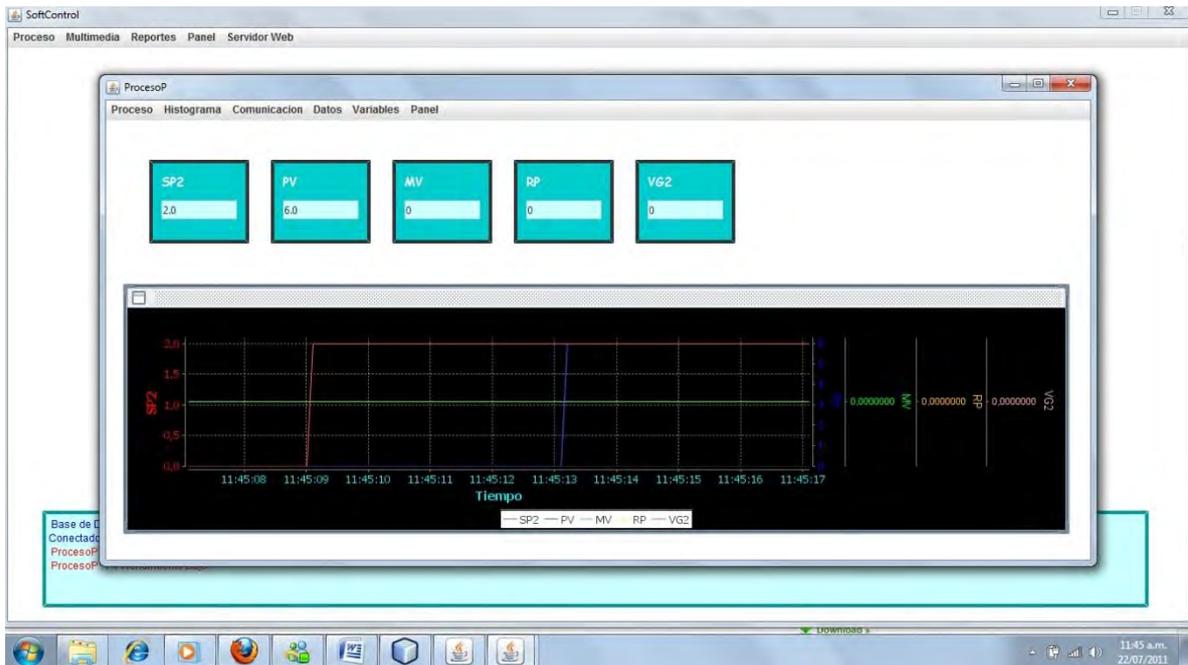


Figura 28 - Proceso recibiendo datos

Menú de almacenamiento de datos del proceso

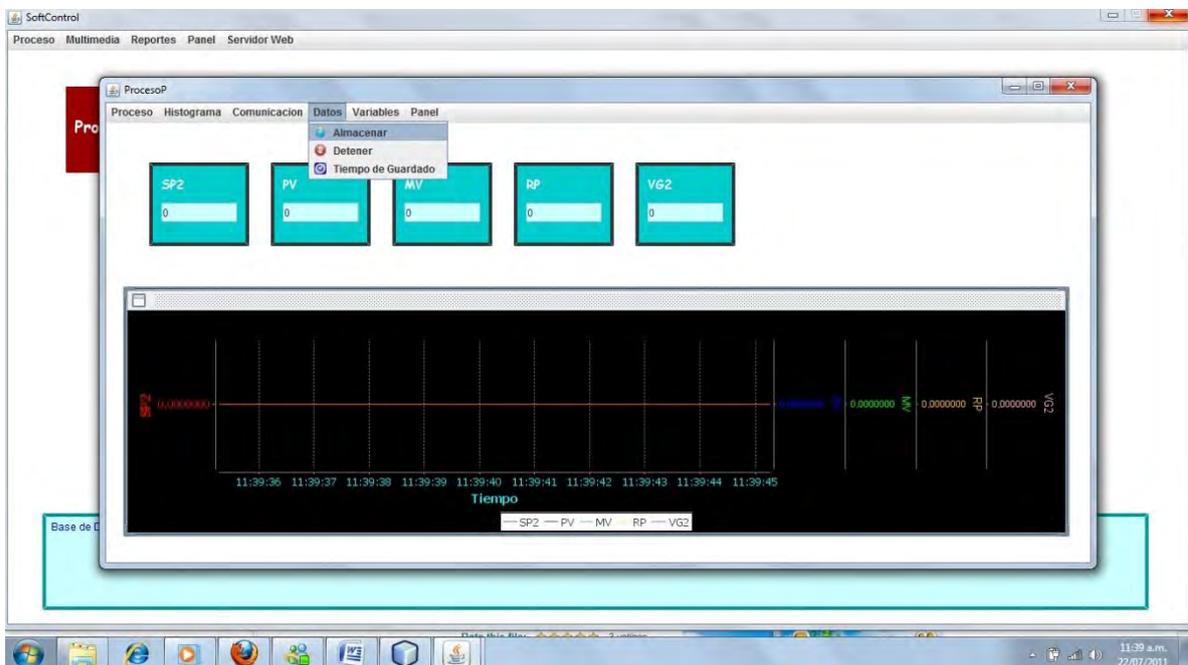


Figura 29 - Opción de guardado de datos

Editando variables del proceso

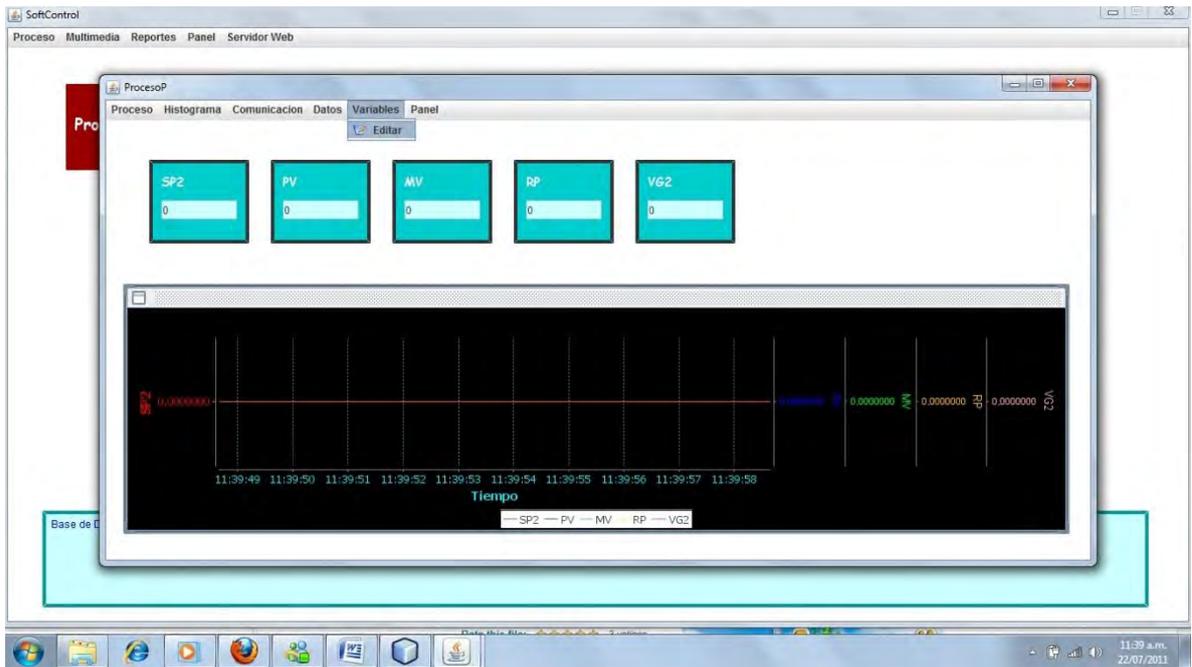


Figura 30 - Opción de edición de variables de proceso

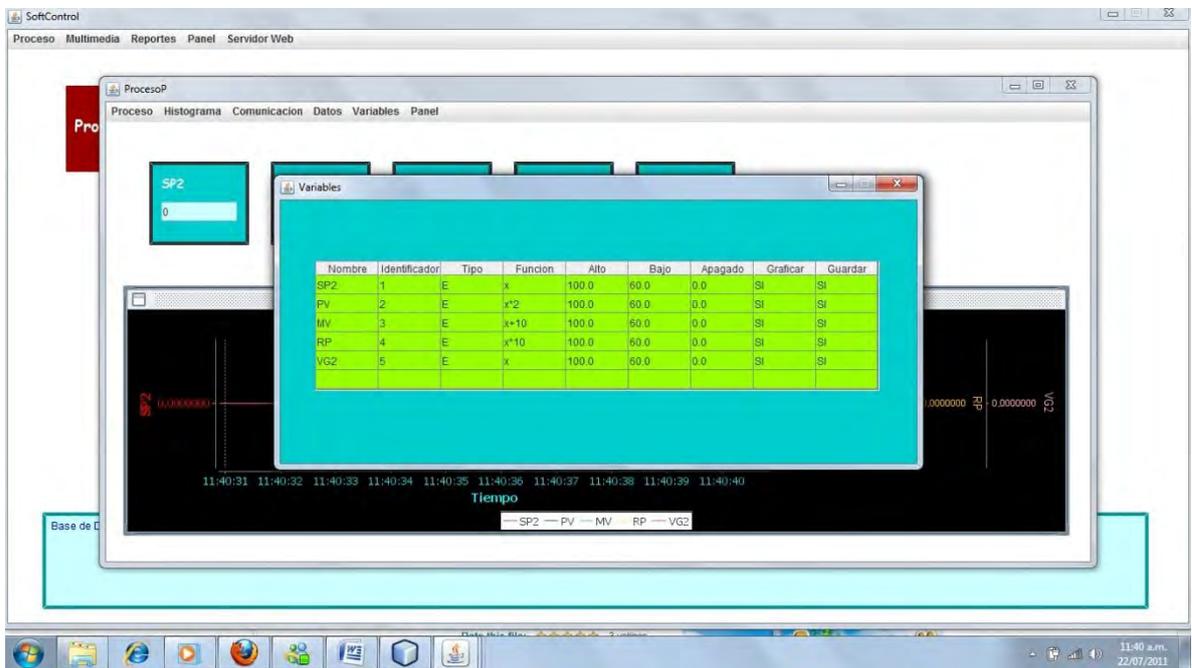


Figura 31 - Variables del proceso

Cambiando color al panel del proceso

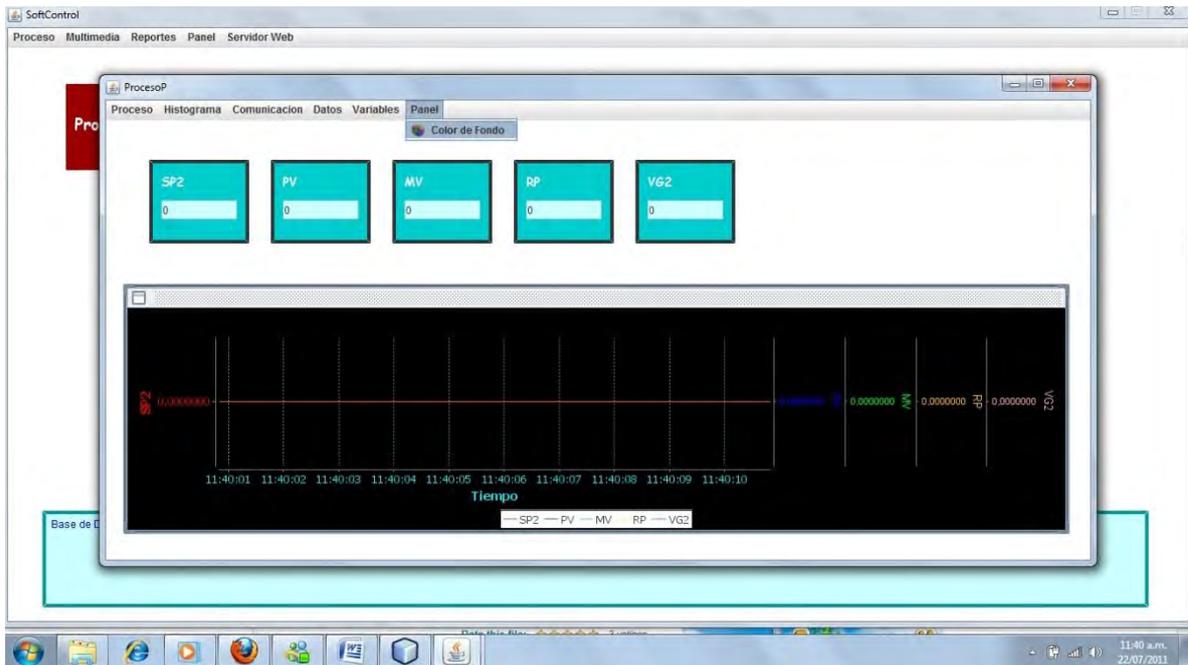


Figura 32 - Opción de cambiar color a panel de proceso

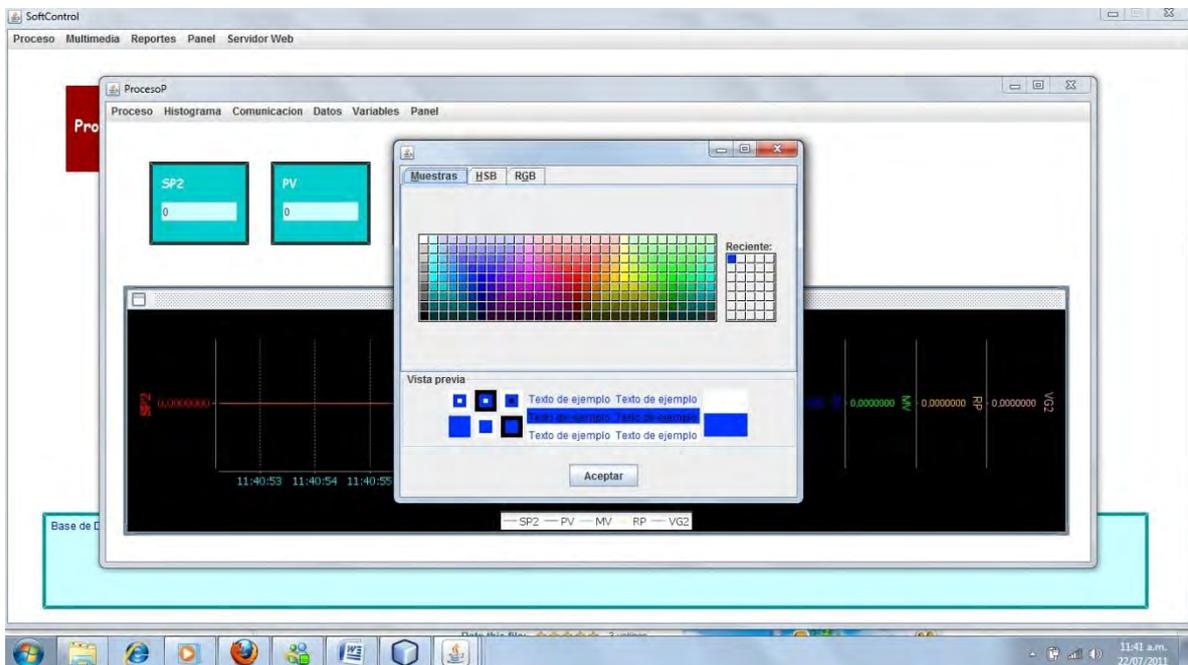


Figura 33 - Seleccionando color para fondo de panel de proceso

Importando imágenes en el panel principal

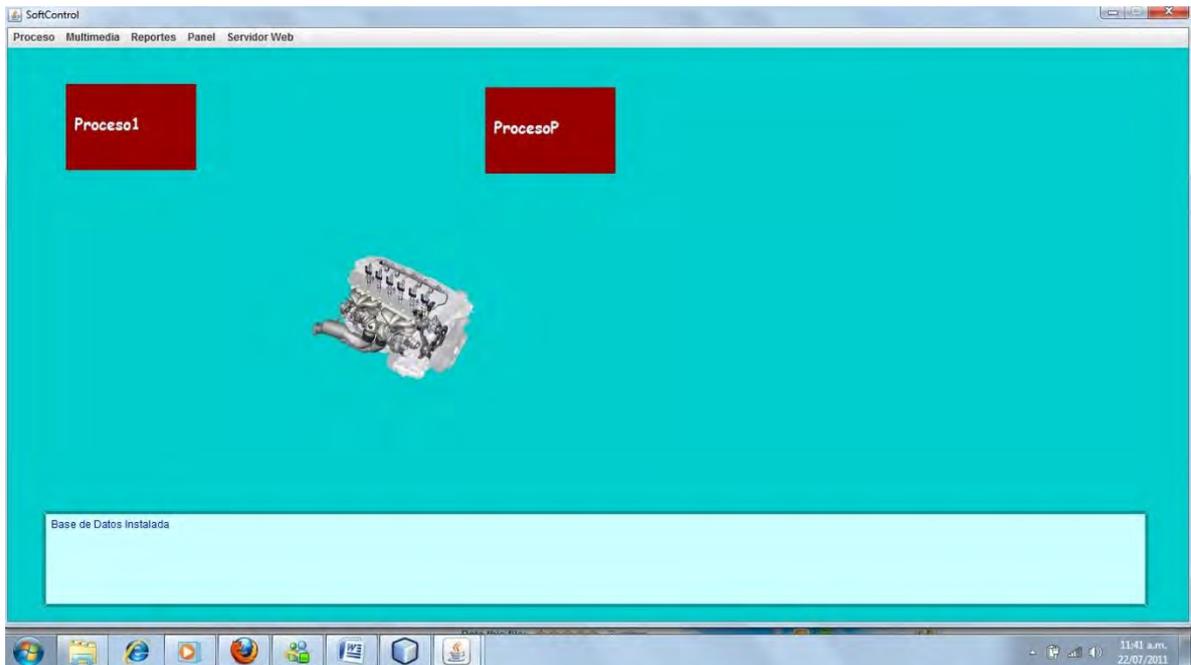


Figura 34 - Importando imágenes

Agregando visualizadores en el panel principal

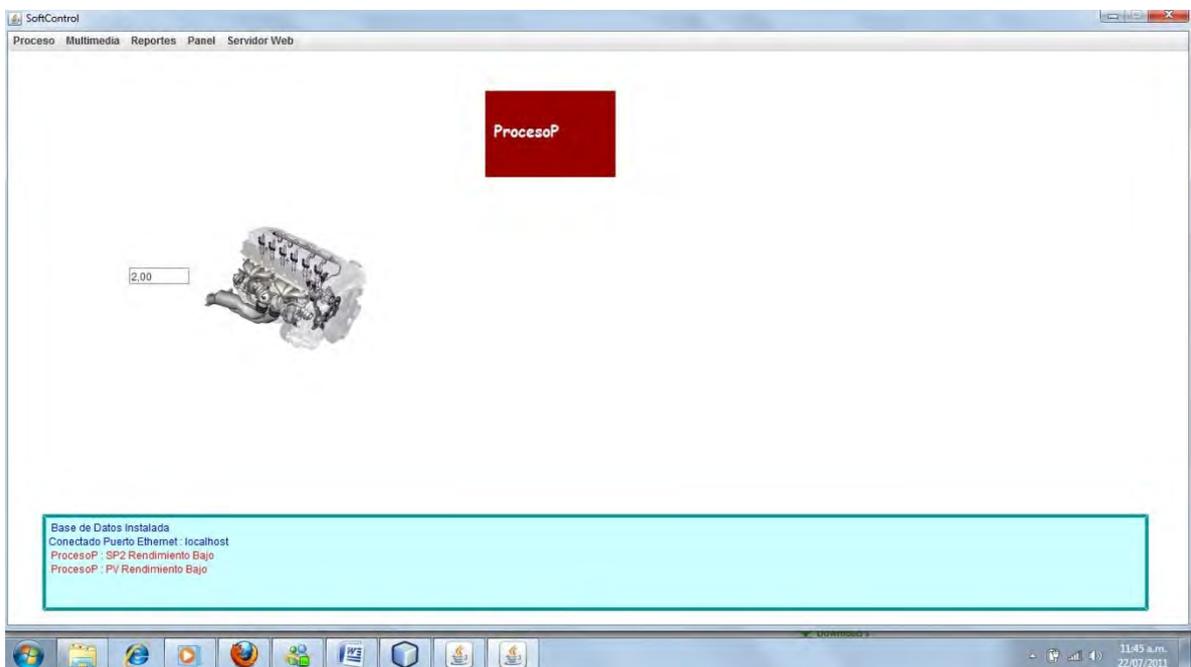


Figura 35 - Agregando visualizadores de variables

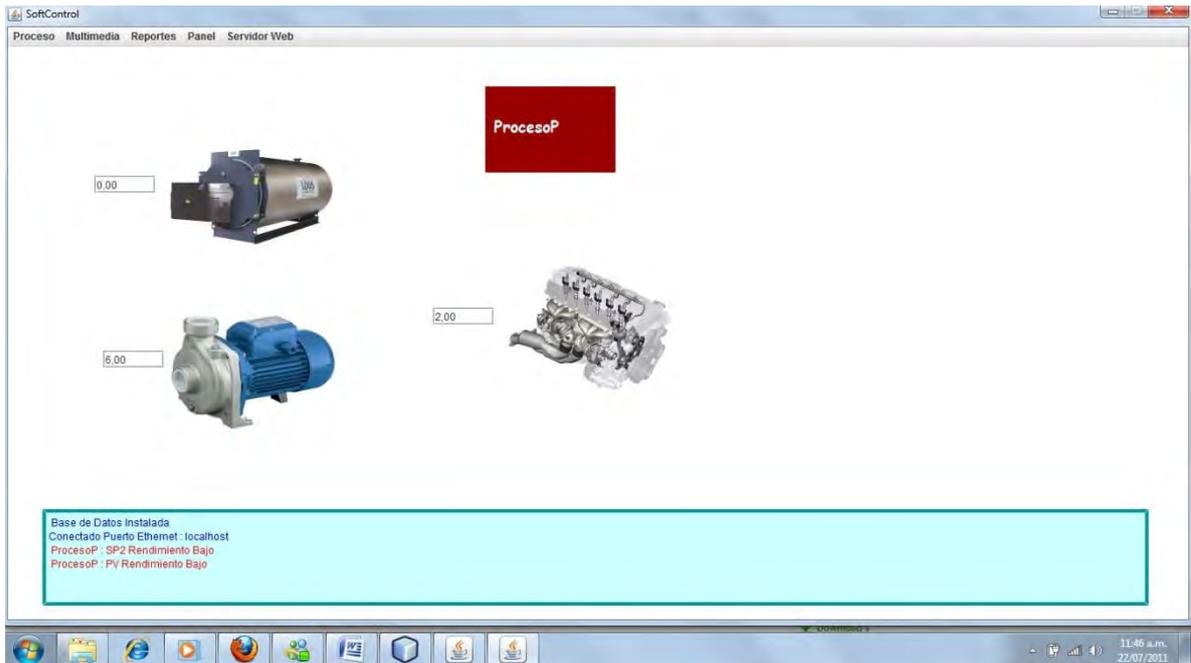


Figura 36 - Visualizadores de variables de proceso

Agregando conectores en el panel principal

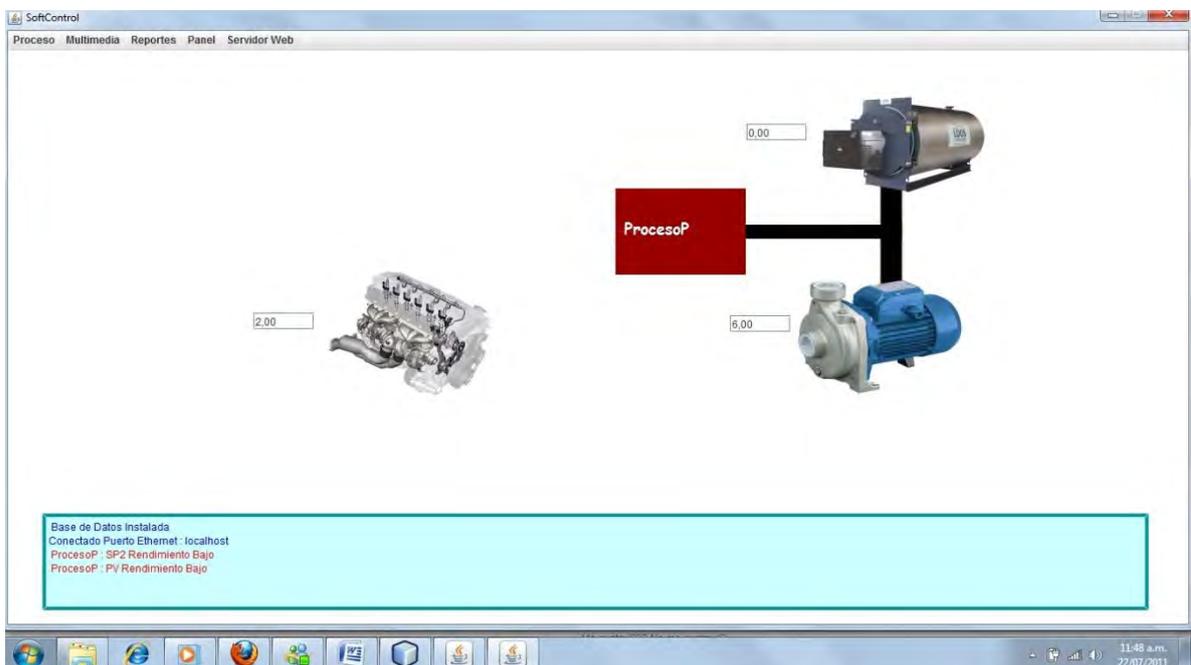


Figura 37 - Agregando conectores

Acceso al sistema web

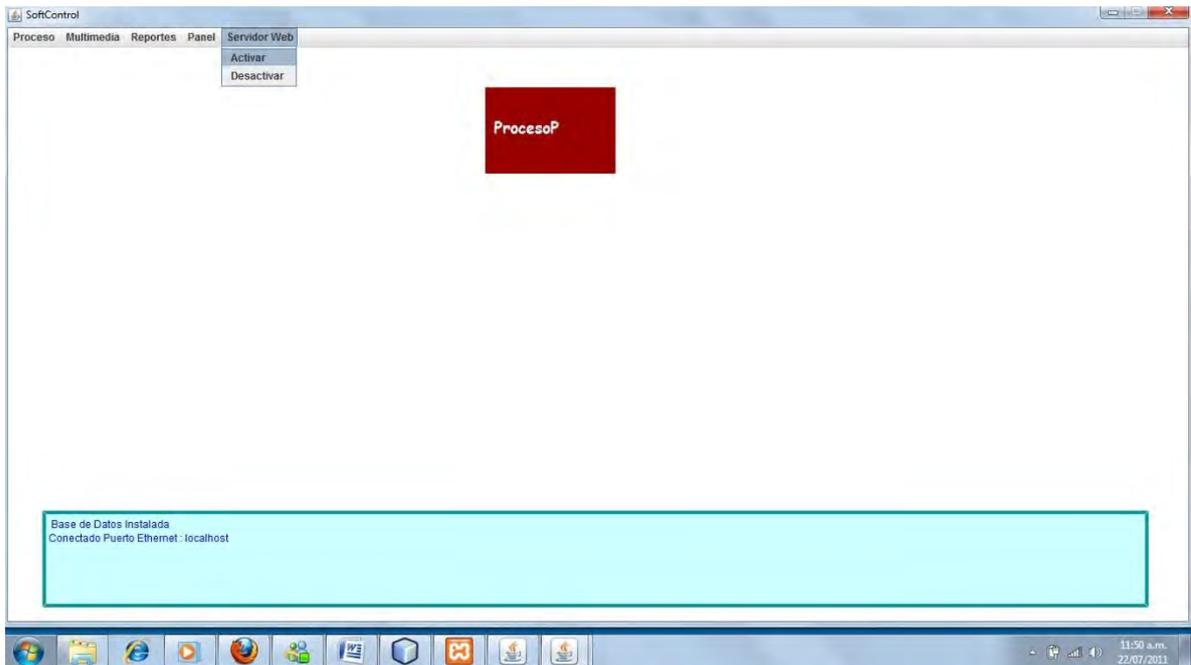


Figura 38 - Activando comunicación web



Figura 39 - Acceso al sistema web

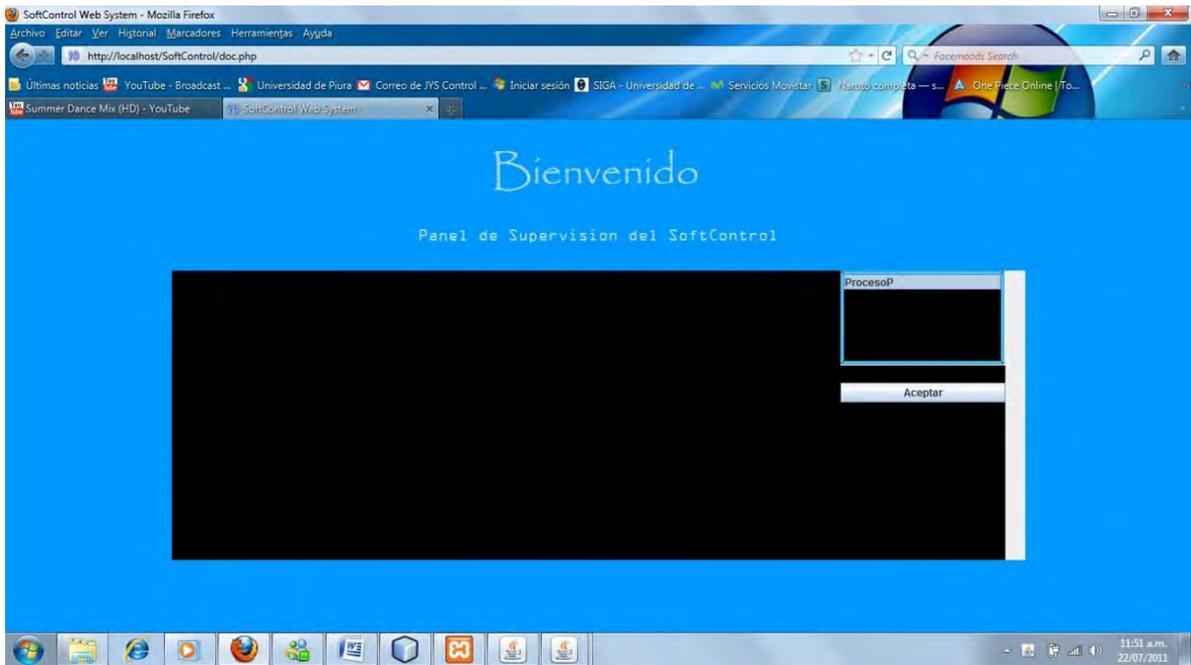


Figura 40 - Seleccionando variables para supervisión



Figura 41 - Supervisión de proceso vía web

2.14 Diseño de la Lógica de Aplicación o de Negocio

2.14.1 Diagrama de Clases de Diseño

La lógica del sistema se representa a continuación a modo de diagrama de clases, además se presentan los atributos de estas y sus métodos.

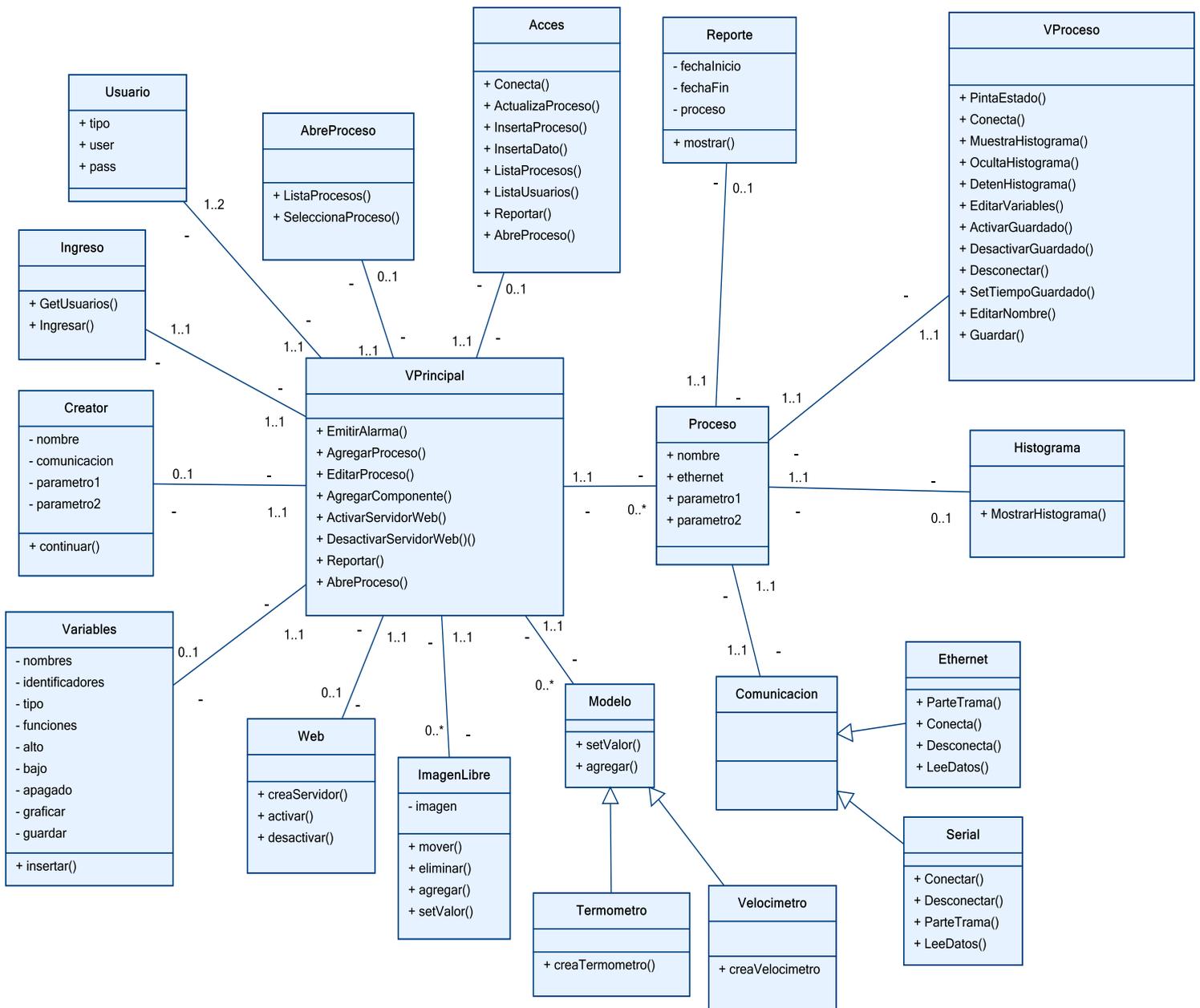


Figura 42 - Diagrama de clases

2.14.2 Detalle del diagrama de clase del diseño

Es importante definir qué rol jugará cada clase en la implementación, que métodos tendrán y como interactuarán entre ellas por ello a continuación se presenta la lista de clases y su respectiva descripción técnica.

Clase VPrincipal:

- EmitirAlarma()

El método EmitirAlarma se encarga de emitir avisos de alarma de estados, conexiones y otros tipos de avisos en el panel principal, este método es invocado por cada subprocesso en el que se ejecutan los procesos dentro del sistema, es decir cada vez que una variable cambie de estado o se pierda la conexión, etc. El hilo correspondiente al proceso llamará dicho método.

- AgregaProceso()

El método AgregaProceso se encarga de inicializar la ventana de la clase Creator, con el cual se podrá crear un nuevo proceso al sistema, dicho método es llamado por un evento de clic en el menú de la ventana principal.

- EditaProceso()

El método EditaProceso se encarga de hacer visible las ventana de los procesos que se encuentran siendo monitoreados desde la ventana Principal, el método EditaProceso es llamado por un evento de clic en los componentes de los respectivos procesos en el panel de VPrincipal.

- AbreProceso()

El método AbreProceso se encarga de Inicializar la ventana AbreProceso para la inicialización de un proceso previamente guardado en la base de datos, dicho método es llamado por un evento de clic en el menú de la clase VPrincipal.

- AgregarComponente()

El método AgregarComponente es el método que inserta las imágenes libres, modelos, visualizadores y procesos al panel principal de la clase VPrincipal, dicho método es llamado por eventos de clic en los componentes del menú en la clase VPrincipal.

- ActivarServidorWeb()

El método ActivarServidorWeb inicializa las comunicaciones con la web y espera peticiones del navegador web para la supervisión vía web, dicho método es llamado por un evento de clic en el menú en la clase VPrincipal.

- DesactivarServidorWeb()

El método DesactivarServidorWeb detiene las comunicaciones con la web y se encarga de tratar cualquier error de comunicación entre la Web y el programa principal, dicho método es llamado por un evento de clic en el menú en la clase VPrincipal o por un error de comunicación en el subproceso de comunicación web.

- Reportar()

El método reportar se encarga de inicializar la ventana Reporte en el cual se realizará una consulta de datos históricos de la base de datos para así presentar una gráfica del proceso en un intervalo de tiempo elegido por el usuario, dicho método es llamado por un evento de clic en el menú de la ventana VPrincipal.

Clase Creator:

- Continuar()

El método continuar se encarga de inicializar la ventana Variables para proceder con el proceso de creación de procesos, específicamente pasar a la parte en donde se configuran las variables del proceso, el método comprueba primero los valores ingresados sean válidos antes de pasar al siguiente paso, dicho método es llamado por un evento de clic en el botón de la clase Creator.

Clase Variables:

- Insertar()

El método Insertar se encarga de crear un nuevo objeto de la clase Proceso con los datos obtenidos de la clase Creator y Variables, así como inicializar la ventana VProceso de cada proceso insertado en el panel de la clase Principal, dicho método es llamado por un evento de clic en el botón de la clase Variables.

Clase AbreProceso:

- SeleccionaProceso()

El método selecciona proceso se encarga de obtener los datos del proceso seleccionado en la lista de dicha clase y abrirlo con todas sus características previamente guardadas, dicho método es llamado por un evento de clic en el botón de la clase AbreProceso.

- ListaProceso()

El método ListaProceso es el método que es llamado al inicializar la ventana ListaProceso, dicho método obtiene todos los procesos de la base de datos y los coloca en la lista del panel de esa clase, dicho método por el método constructor de ListaProceso al momento de inicializar la ventana.

Clase BaseDatos:

- Conecta()

El método conecta se encarga de establecer la comunicación entre el sistema y su base de datos en el programa Microsoft Access, dicho método es llamado al iniciar el programa, específicamente es llamado en el método constructor de VPrincipal.

- ActualizaProceso()

El método ActualizaProceso se encarga de realizar una sentencia de UPDATE en la base de datos actualizando todos los datos del proceso seleccionado, dicho método es llamado por un evento de clic en el menú de la ventana VProceso, específicamente en el menú guardar.

- InsertaProceso()

El método InsertaPrceso se encarga de realizar la sentencia INSERT en la base de datos, es decir se encarga de ingresarlos a la base de datos, obviamente antes de llegar a dicho método la información del proceso es verificada para que su registro sea válido, dicho método es llamado al igual que el método ActualizaProceso por un evento de clic en el menú guardar de la clase VProceso.

- InsertaDato()

El método InsertaDato se encarga de guardar los datos relacionados a las variables de los procesos en la base de datos, dicho método es llamado por un contador de tiempo de cada proceso al cual el usuario definió previamente el intervalo de guardado.

- ListaProceso()

El método ListaProceso se encarga de obtener la lista de procesos guardados en la base de datos, dicho método es llamado por el método ListaProceso de la clase AbreProceso.

- ListaUsuarios()

El método ListaUsuarios se encarga de obtener una lista de los usuarios registrados del sistema desde la base de datos para así poder compararlos con los datos ingresados en la clase Ingresar y el usuario pueda tener acceso al sistema, dicho método es llamado por el método GetUsuarios de la clase Ingresar.

- Reportar()

El método Reportar realiza una consulta a la base de datos pidiendo los registros de los datos de un proceso determinado en un periodo de tiempo determinado, dicho método es llamado por el método Reportar de la clase Reporte.

- AbreProceso ()

El método abre proceso hace una búsqueda en la base de datos de un proceso determinado y extra sus datos para luego ser utilizados en crear un objeto de la clase Proceso y ser agregado en el panel de VPrincipal, dicho método es llamado por el evento de clic del botón de la clase AbreProceso.

Clase Ingreso:

- GetUsuarios()

El método GetUsuarios se encarga de llamar al método ListaUsuarios para obtener los usuarios existentes del sistema y así poder comparar la información que ingresará el usuario, dicho método es llamado por el constructor de la clase Ingreso.

- Ingresar()

El método Ingresar se encarga de inicializar la ventana VPrincipal solo si los datos ingresados por el usuario coinciden con algún registro de usuarios existente en la base de datos, dicho método es llamado por un evento de clic en el botón de la clase Ingreso.

Clase Reporte:

- Mostrar()

El método Mostrar se encarga de crear el gráfico donde se mostrarán los datos históricos del reporte y además de llamar al método Reportar de la clase BaseDatos, dicho método es llamado en el constructor de la clase Reporte, es decir al inicializarse la ventana.

Clase VProceso:

- PintaEstado()

El método PintaEstado se encarga de examinar si hay un cambio de estado en una variable cada vez que un dato es recibido y muestra una alarma en el panel principal de VPrincipal, dicho método es llamado en el subproceso de comunicación de cada proceso.

- Conecta()

El método Conecta se encarga de llamar a las clases de comunicación respectiva (Serial o Cliente) según sea el caso para lograr conectar el proceso con los dispositivos de trabajo, el método Conecta es llamado por un evento de clic en el botón conectar de la ventana VProceso.

- MuestraHistograma()

El método MuestraHistograma se encarga de llamar al constructor de la clase Histograma para poder así crear el histograma en tiempo real y poder monitorear las variables del proceso, dicho método es llamado por un evento de clic en el menú de histograma de la ventana VProceso.

- OcultaHistograma()

El método OcultaHistograma se encarga de esconder de la pantalla el histograma del proceso hasta que el usuario decida mostrarlo nuevamente, dicho método es llamado por un evento de clic en el menú de histograma de la ventana VProceso.

- DetenHistograma()

El método DetenHistograma se encarga de detener el contador de tiempo que agrega nuevos datos al histograma, por lo tanto el histograma se quedaría estático hasta que el usuario decida continuarlo, dicho método es llamado por un evento de clic en el menú de histograma de la ventana VProceso.

- EditaVariables()

El método EditaVariables se encarga de abrir la ventana Variables de cada proceso para poder editar los parámetros y los identificadores de comunicación, dicho método es llamado por un evento de clic en el menú de la clase VProceso.

- ActivarGuardado()

El método ActivarGuardado se encarga de empezar el contador de tiempo para el guardado de los datos hasta que el usuario decida detenerlo, dicho método es llamado por un evento de clic en el menú de guardado de datos en la clase VProceso.

- DesactivarGuardado()

El método DesactivarGuardado detiene el contador de tiempo que se encarga de guardar los datos de las variables hasta que el usuario desee continuar almacenando, dicho método es llamado por un evento de clic en el menú desactivar en la clase VProceso.

- Desconectar()

Se encarga de detener el subprocesso de comunicación entre el proceso y los dispositivos hasta que el usuario desee continuar la comunicación, dicho método es llamado por un evento de clic en el menú desconectar en la clase VProceso.

- SetTiempoGuardado()

El método SetTiempoGuardado se encarga de pedir al usuario que ingrese un periodo de tiempo, el cual se usará como intervalo de guardado de los datos tomados de la comunicación y serán guardados en la base de datos, el método es llamado por un evento de clic en el menú de tiempo de guardado en la clase VProceso.

- EditarNombre()

El método EditarNombre permite al usuario ingresar un nuevo nombre al proceso, para que se guarde el nuevo nombre hay que hacer clic en guardar después de editar el nombre, el método EditarNombre es llamado por un evento de clic en el menú de editar nombre de la clase VProceso.

- Guardar()

El método Guardar se encarga de guardar la información del proceso en la base de datos, incluyendo el nombre, las variables, identificadores y todas las configuraciones realizadas al mismo, dicho método es llamado por un evento de clic en el menú guardar de la clase VPrincipal.

Clase Histograma

- MostrarHistograma()

El método MostrarHistograma se encarga de preparar el gráfico ya sea el histograma para los datos en tiempo real o el histograma para los reportes, según sea el tipo de orden del objeto de dicha clase, el histograma como reporte se pone en el panel de dicha clase para mostrarlo al usuario. El método MostrarHistograma es llamado por un evento de clic en el menú de mostrar histograma en VProceso o por un evento de clic en el botón de aceptar de la clase Reporte.

Clase Web

- CreaServidor()

Se encarga de crear un subproceso de comunicación para la página web y el servidor, una vez lista la comunicación ya puede aceptar usuarios web, dicho método es llamado en el método constructor de la clase Web, la cual se inicializa al inicializarse VPrincipal.

- Activar()

Se encarga de establecer la comunicación web, empezar a recibir datos, interpretarlos e interactuar con la web en tiempo real, dicho método es llamado por un evento de clic en el menú activar web de la clase VPrincipal.

- Desactivar()

Se encarga de detener el subproceso de comunicación web cuando se requiera hasta que el usuario desee continuarlo, dicho método se activa con un evento de clic en el menú desactivar web de la clase VPrincipal.

Clase ImagenLibre

- Mover()

Se encarga de colocar el objeto de la clase ImagenLibre en la posición que se encuentra el mouse dentro del panel de VPrincipal, dicho método es llamado por un evento de arrastre de ratón dentro del panel de VPrincipal.

- Eliminar()

El método Eliminar se encarga de eliminar la imagen libre del panel VPrincipal en el caso que así se desee, dicho método es llamado por un evento de clic en el menú eliminar del submenú del objeto ImagenLibre.

- Agregar()

El método Agregar se encarga de colocar la imagen libre en el panel de VPrincipal, dicho método es llamado al crear el objeto ImagenLibre.

- SetValor()

El método SetValor se encarga de darle el valor de la variable que representa la ImagenLibre en caso de ser del tipo de visualizador para poder así monitorear una variable desde el panel principal de la clase VPrincipal, dicho método es llamado cada vez que se recibe un dato en el subproceso de comunicación de la respectiva variable.

Clase Modelo

- SetValor()

El método SetValor se encarga de visualizar el valor asociado a la variable del modelo correspondiente, dicho método es llamado cada vez que se recibe un nuevo dato de la variable representada.

- Agregar()

El método Agregar se encarga de colocar el objeto Modelo al panel principal de la clase VPrincipal para así tener una mejor supervisión del proceso sin tener que abrirlo y ver el histograma, el método es llamado al inicializarse el objeto Modelo.

Clase Termómetro

- CreaTermómetro()

Se encarga de crear el modelo termómetro con sus parámetros de valor mínimo y máximo, dicho método es llamado al inicializarse el objeto.

Clase Velocímetro

- CreaVelocimetro()

Se encarga de crear el modelo velocímetro con sus parámetros de valor mínimo y máximo, dicho método es llamado al inicializarse el objeto.

Clase Ethernet

- ParteTrama()

Se encarga de separar el identificador del valor de la variable recibida para así asignarla a la variable sensada correspondiente, dicho método es llamado al recibir una trama completa en el subproceso de comunicación correspondiente del proceso.

- Conecta()

Se encarga de establecer la comunicación mediante el protocolo Ethernet entre un proceso y un dispositivo conectado al servidor, dicho método es llamado por el método conectar de la clase VProceso.

- Desconecta()

Se encarga de detener el proceso de comunicación entre el dispositivo conectado y el proceso, el método es llamado por el método desconectar de la clase VProceso o por un evento de error en el subproceso de comunicación.

- LeeDatos()

Se encarga de inicializar el subproceso de comunicación por el protocolo Ethernet, recibiendo datos y analizándolos con el método ParteTrama, dicho método es llamado por el método Conecta de la clase Ethernet.

Clase Serial

- Conectar()

Se encarga de establecer la comunicación mediante el protocolo Serial entre un proceso y un dispositivo conectado al servidor, dicho método es llamado por el método conectar de la clase VProceso.

- Desconectar()

Se encarga de detener el proceso de comunicación entre el dispositivo conectado y el proceso, el método es llamado por el método desconectar de la clase VProceso o por un evento de error en el subproceso de comunicación.

- ParteTrama()

Se encarga de separar el identificador del valor de la variable recibida para así asignarla a la variable sensada correspondiente, dicho método es llamado al recibir una trama completa en el subproceso de comunicación correspondiente del proceso.

- LeeDatos()

Se encarga de inicializar el subproceso de comunicación por el protocolo Serial, recibiendo datos y analizándolos con el método ParteTrama, dicho método es llamado por el método Conecta de la clase Serial.

Capítulo 3: Implementación

Durante el desarrollo del sistema SCADA se emplearon diferentes tecnologías, muchas veces se tuvo que hacerlas interactuar y comunicarlas para así lograr los objetivos. Por esa razón la implementación presenta otro capítulo en el presente documento. La implementación es otra parte del “Know-How” del sistema en conjunto.

3.1 Metodología de desarrollo:

3.1.1 Metodologías actuales

Actualmente existen numerosas metodologías de desarrollo de software. Tenemos las tradicionales que se centran especialmente en el control del proceso pero toman mayor tiempo y tenemos las metodologías ágiles, las cuales dan mayor valor a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas es decir el desarrollo en menor tiempo.

Tabla 1 - Comparativa entre metodologías actuales y tradicionales (Letelier, 2006)

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparados para cambios durante el proyecto.	Cierta resistencia a los cambios.
Proceso menos controlado, con pocos principios.	Proceso mucho más controlado, con numerosas normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.

El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

3.1.2 Selección de la metodología

Existen varias metodologías tanto tradicionales como ágiles, entre las más famosas encontramos: Proceso Unificado de Rational (RUP), *Scrum*, *Xtreme Programming* (XP), Proceso unificado ágil (AUP), *Microsoft Solution Framework* (MSF).

A continuación se presentará una tabla comparativa que resume las características de dichas metodologías:

Tabla 2 - Comparativa metodologías de desarrollo (Letelier, 2006)

Metodología	Contacto con Usuario	Documentación requerida	Requerimiento de Personal	Velocidad de desarrollo
RUP	Bajo	Alta	Cualificado	Normal
SCRUM	Muy Alto	Poca	Cualificado	Normal
XP	Alto	Poca	Cualificado	Muy Alta
AUP	Medio	Media	Muy Cualificado	Alta
MSF	Medio	Demasiada	Cualificado	Normal

Cabe mencionar que no existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc).

La metodología elegida es la XP pues ofrece la mayor cantidad de información para el diseño, el menor tiempo de desarrollo y que según la investigación realizada es la más popular entre los desarrolladores hoy en día.

3.2 Librerías Usadas

Para la implementación del sistema se usaron diversas librerías externas al API nativo de JAVA, las cuales cumplen determinadas funciones dentro del sistema, dichas librerías son las siguientes:

JFree Chart

Esta librería se usó en la parte de control y monitoreo de los procesos en planta, al igual que se usó para presentar los reportes históricos.

Es una librería de código libre para programación en lenguaje JAVA, con ésta librería es muy sencillo realizar gráficos, cuadros en las aplicaciones, dicha librería posee una amplia documentación de sus métodos y de la construcción de todos los tipos de gráficos. La librería puede crear no solo gráficos en 2D sino también en 3D, gráficos de barra, líneas de tiempo, gráficos de pronósticos, etc.

La ventaja de su uso es que la librería es “*open source*” lo que quiere decir que es gratis, además dispone de una fácil integración con el IDE Netbeans lo cual facilita el trabajo (JFree, 2012).



Figura 1 - Ejemplo de gráfico de pie (JFree, 2012)

Giovynt Serial Port

Esta librería nos dio la posibilidad de usar la comunicación serial para nuestro sistema, gracias a ella dispositivos que usan este tipo de comunicación son compatibles con el sistema.

Giovynt Serial Port es una librería que permite usar el lenguaje de programación java para crear aplicaciones de sistemas de control y adquisición de datos, implementa métodos de envío y recepción de datos usando el sistema de caracteres ASCII mediante la comunicación RS-232, también soporta la comunicación por medio de cable conversor Serial-Usb (Giovynt, 2012).



Figura 2 - Librería para comunicación serial (Giovynet, 2012)

Math Evaluator

Esta librería le dio una funcionalidad más al sistema, además de darle flexibilidad en su operación, la librería permitía operar los datos recibidos directamente de los dispositivos, pasarla por una expresión matemática y obtener el resultado.

Evaluable de expresiones matemáticas, una librería muy potente que permite tratar los datos de manera que no se necesite reconfigurar el dato enviado desde el dispositivo, sino que permite al sistema una vez recibido el dato modificarlo según convenga, es decir opera aritméticamente el dato recibido o antes de salir por medio de la expresión configurada por el usuario. (Code Google, 2011)

DateChooser

Esta librería es perfecta para seleccionar datos de tipo fecha en el sistema, se usó en la parte de selección del rango de fechas para el reporte histórico.

DateChooser es una librería para lenguaje java con componentes de la librería swing, contiene 3 componentes para la implementación: Un panel, un editor de fechas y una ventana de diálogo, los cuales son totalmente compatibles con cualquier IDE de programación. (JDateChooser, 2012)

Dicha librería permite principalmente:

- Seleccionar fechas, periodos, horas, etc.
- Permite configurar su apariencia dentro de la ventana
- Editar las celdas de información

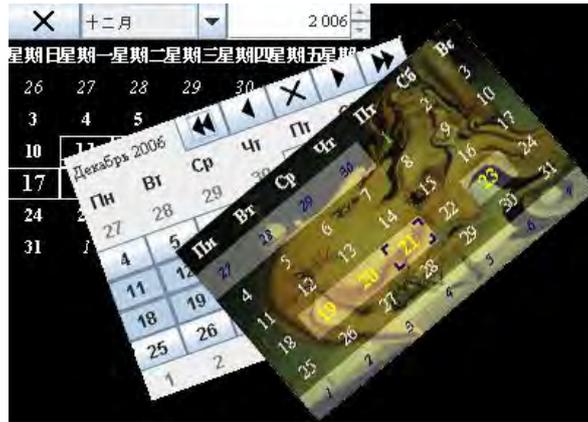


Figura 3 - Librería para seleccionar fechas (JDateChooser, 2012)

3.3 Seguridad

3.3.1 Encriptación MD5

El sistema usa el algoritmo MD5 (abreviatura de *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5) tanto para convertir claves del sistema web como para almacenar datos del sistema en planta, tanto PHP como java implementan este algoritmo en su API nativo.

MD5 es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (*Massachusetts Institute of Technology*, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4 después de que Hans Dobbertin descubriese su debilidad

La codificación del MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal.

A pesar de haber sido considerado criptográficamente seguro en un principio, ciertas investigaciones han revelado vulnerabilidades que hacen cuestionable el uso futuro del MD5. En agosto del 2004, Xiaoyun Wang, Dengguo Feng, Xuejia Lai y Hongbo Yu anunciaron el descubrimiento de colisiones de hash para MD5, además aseguraron que una clave de 128 bits no era tan difícil de atacar por fuerza bruta (iterar claves aleatorias hasta encontrar la verdadera).

Cabe resaltar que MD5 es un algoritmo muy seguro, pero definitivamente al avanzar la capacidad de hardware en el día a día, todos los algoritmos serán más fácilmente vulnerados por un ataque de fuerza bruta (Scribd, 2013)

3.3.2 Encriptación MySQL

Otra alternativa de encriptación de datos es la encriptación AES en MySQL y se realiza mediante dos funciones: `AES_ENCRYPT` y `AES_DECRYPT`. (Rock, 2012)

Su uso es muy sencillo:

Una inserción de un registro encriptado se realizaría de la siguiente forma:

- `insert into tb_ejemplo values (aes_encrypt ('valor a encriptar como cadena','contraseña'));`

Para obtener los datos realizamos un select normal de la siguiente forma:

- `select aes_decrypt (columna_encriptada, 'contraseña') from tb_ejemplo;`

3.3.3 Protocolo de comunicación propio

El sistema, motivo de la presente tesis, cuenta con su propio protocolo de comunicación, es un protocolo de mayor nivel desarrollado en base a los protocolos de comunicación mencionados anteriormente. Muchos SCADA comerciales y de código abierto funcionan con el envío de información continua facilitando la interpretación de datos a los agentes indeseables. “Soft-Control” maneja un sistema de identificadores que cambian según los requerimientos del usuario, con esto tendríamos un identificador para cada variable y en un determinado lapso de tiempo, con lo cual la información no se interpretará tan fácilmente por agentes desconocidos.

Conclusiones

- Los sistemas SCADA están evolucionando conjuntamente con las tecnologías de la información, esto implica que el hardware ya no es limitante del mismo. Es el software el que le da el mayor valor agregado al sistema.
- Las tecnologías de código abierto son igual de competentes e incluso más competentes que algunas tecnologías de pago. Esto significa que las tecnologías de código libre son una tendencia para la industria del software en el futuro.
- Actualmente, el mayor limitante de los sistemas SCADA no se encuentra alojado en alguna característica del hardware o en el software. Las comunicaciones son un tema muy delicado pues se debe manejar de tal modo que la seguridad no sea baja, ni que baje las velocidades de transferencia de datos.
- Los gestores de bases de datos ciertamente son de gran utilidad hoy en día en el desarrollo de estos tipos de software, pues ellos se encargan de funciones tales como la seguridad, almacenamiento, reportes, alertas, etc. Dejando así al sistema SCADA más recursos para que cumpla con sus funciones específicas de supervisión, control y adquisición de datos.
- Las metodologías tanto de diseño como de desarrollo de software también son un punto a resaltar hoy en día. No solo se pueden aplicar en el campo de la informática, sino que también, como el caso de la presente tesis, se pueden aplicar para proyectos industriales tales como el desarrollo o implementación de sistemas SCADA.
- Con los sistemas de código abierto la pequeña y mediana empresa tiene una gran oportunidad de volverse más competitivo, tener mayor control sobre sus procesos, agilizar el manejo de la información y poder llegar a más clientes.
- Con el software de código libre, la pequeña y mediana empresa puede implementar un centro de desarrollo de software. Logrando así disminuir sus costos en software de pago y tener un software hecho a la medida que se adapte totalmente a su forma de operar.

Glosario

Término	Significado
BD	Base de datos
GUI	Interfaz gráfica de usuario
LN	Lógica de negocios
SQL	Lenguaje Estructurado de Consultas
HILO	Subproceso del sistema, en programación se puede dar piezas de código que se ejecutan en paralelo a otras
PLC	Controlador lógico programable
OPC	Ole for process control
MD5	Algoritmo de Resumen del Mensaje 5
UML	Lenguaje Unificado de Modelado

Tabla 1 - Glosario del documento

Bibliografía

- Academic. (20 de Febrero de 2012). *Puerto serie*. Recuperado el 20 de Febrero de 2012, de Puerto serie: <http://www.esacademic.com/dic.nsf/eswiki/971671>
- Academic. (20 de Febrero de 2013). *Tarjeta de red*. Recuperado el 20 de Febrero de 2013, de Tarjeta de red: <http://www.esacademic.com/dic.nsf/eswiki/37121>
- Alegsa. (12 de Noviembre de 2010). *Ventajas y desventajas del modelo cliente-servidor*. Recuperado el 13 de Enero de 2013, de Ventajas y desventajas del modelo cliente-servidor: http://www.alegsa.com.ar/Respuesta/ventajas_y_desventajas_del_modelo_clienteservidor.htm
- Blogspot. (20 de Febrero de 2012). *Equipos Informáticos*. Recuperado el 20 de Febrero de 2012, de Equipos Informáticos: <http://grupo3-equipos-informaticos.blogspot.com/2011/03/rj45.html>
- Breeding Designers. (29 de Marzo de 2009). *Servidor web*. Recuperado el 03 de Enero de 2013, de Servidor web: <http://www.breedingdesigners.com.ar/articulos/servidorweb.html>
- Code Google. (04 de Febrero de 2011). *Graphing Calculator*. Recuperado el 03 de Diciembre de 2012, de Graphing Calculator: <http://code.google.com/p/graphingcalculator/source/browse/Calculator/src/expressions/MathEvaluator.java?r=121>
- Corrales, L. (Diciembre de 2007). Recuperado el 3 de Diciembre de 2012, de <http://bibdigital.epn.edu.ec/bitstream/15000/10020/2/PARTE%202.pdf>
- Culturación. (20 de Febrero de 2012). *Convertir un PC en servidor web*. Recuperado el 20 de Febrero de 2012, de Convertir un PC en servidor web: <http://culturacion.com/2011/12/como-convertir-un-pc-en-servidor-web/>
- Diaz, H. M. (03 de Diciembre de 2012). Obtenido de <http://www.galeon.com/hamd/pdf/scada.pdf>
- García, J. (10 de Diciembre de 2012). *HISTORIA SCADA*. Recuperado el 10 de Diciembre de 2012, de HISTORIA SCADA: <http://es.scribd.com/doc/76466757/HISTORIA-SCADA>

- Geocities. (11 de Diciembre de 2012). *FUNCIONES DEL SISTEMA SCADA*. Recuperado el 11 de Diciembre de 2012, de FUNCIONES DEL SISTEMA SCADA:
http://www.oocities.org/gabrielordonez_ve/FUNCIONES_DEL_SISTEMA_SCADA.htm
- Giovynet. (03 de Diciembre de 2012). Recuperado el 03 de Diciembre de 2012, de
http://www.giovynet.com/index_en.html
- Java. (13 de Enero de 2013). *¿Qué es la tecnología Java y por qué lo necesito?* Recuperado el 13 de Enero de 2013, de ¿Qué es la tecnología Java y por qué lo necesito?:
http://www.java.com/es/download/faq/whatis_java.xml
- JDateChooser. (03 de Diciembre de 2012). Recuperado el 03 de Diciembre de 2012, de
<http://jdatechooser.sourceforge.net/>
- JFree. (11 de Setiembre de 2012). *JFreeChart*. Recuperado el 03 de Diciembre de 2012, de JFreeChart: <http://www.jfree.org/index.html>
- Lenguajes de Programación. (03 de Diciembre de 2012). Recuperado el 03 de Diciembre de 2012, de <http://www.lenguajes-de-programacion.com/programacion-java.shtml>
- Letelier, P. (15 de Enero de 2006). *Métodologías ágiles para el desarrollo de software*. Recuperado el 09 de Febrero de 2013, de Metodologías ágiles para el desarrollo de software: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Lozano, C. d. (11 de Diciembre de 2012). *Introducción a SCADA*. Recuperado el 11 de Diciembre de 2012, de Introducción a SCADA:
<http://www.uco.es/investiga/grupos/eatco/automatica/ihm/descargar/scada.pdf>
- Oness. (20 de Febrero de 2012). *Aplicaciones en capas*. Recuperado el 20 de Febrero de 2012, de Aplicaciones en capas: <http://oness.sourceforge.net/proyecto/html/ch03s02.html>
- Pérez, J. M. (2007). *¿Qué es MySQL?* Recuperado el 13 de Enero de 2013, de ¿Qué es MySQL?:
<http://www.espestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm>
- Pineda, F. (11 de Diciembre de 2012). *Sistemas SCADA características y funcionalidades*. Recuperado el 11 de Diciembre de 2012, de Sistemas SCADA características y funcionalidades: <http://es.scribd.com/doc/18188206/Sistema-SCADA>
- Rock, D. (10 de Setiembre de 2012). *Encriptación AES en MySQL*. Recuperado el 17 de Enero de 2013, de Encriptación AES en MySQL:
<http://donnierock.wordpress.com/2012/09/10/encrptacion-aes-en-mysql-y-mariadb/>
- Rodriguez Penin, A. (23 de Diciembre de 2012). *Sistemas SCADA*. Recuperado el 23 de Diciembre de 2012, de Sistemas SCADA:
<http://books.google.com.pe/books?id=cNQfjbBcUq8C&printsec=frontcover&hl=es#v=onepage&q&f=false>
- Scribd. (17 de Enero de 2013). *Algoritmo MD5*. Recuperado el 17 de Enero de 2013, de Algoritmo MD5: <http://es.scribd.com/doc/7905830/Algoritmo-MD5>

Udep. (2008). *Informe Técnico "Desarrollo e investigación de embebidos para aplicación de automatización y control de bajo costo para PYMES.* Piura.

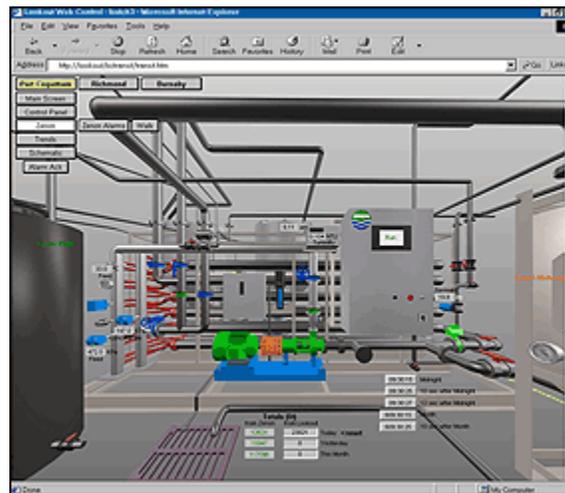
Udep. (2010). *Informe de comparación SCADAS Comerciales.* Piura.

ANEXOS

A-1 Descripción detallada de SCADAS

A continuación se presentan los SCADAS más conocidos en la actualidad, se describen sus capacidades, beneficios, inconvenientes y algunas particularidades únicas de cada sistema (Udep, 2008).

LookOut (National instrument)



Conectividad Abierta

- Capacidad OPC para cliente y servidor
- Enlace a múltiples objetos en la interface HMI de LookOut
- Conexión a ambientes de programación tales como Visual Basic, Java, C++, etc.
- Conexión a una amplia gama de instrumentos de National Instrument.

Configuración en línea

- Las aplicaciones pueden ser modificadas sin interrumpir la comunicación con los dispositivos.

- Elimina vacíos en los datos y monitorea alarmas.
- Desarrolla y modifica aplicaciones sin entrar a inactividad al proceso

Registros distribuidos

- Registra y recupera datos desde cualquier computador en la red.
- Usa herramientas basadas en configuración de registros para el registro instantáneo de datos.
- Mantenimiento exacto de los históricos del proceso y su mantenimiento

Alarmas y eventos distribuidos

- Visualización y reconocimiento de alarmas y eventos desde cualquier parte en la red
- Informa a celulares cuando se activa una alarma.
- Usa un visualizador de datos históricos para mejorar la visualización y administración de los datos.

Seguridad

- Bloqueo del sistema para que los usuarios no puedan modificar el sistema una vez entre en funcionamiento.
- Provee niveles de lectura y escritura a procesos, folders y objetos basados en niveles de seguridad.
- Activación de seguridad basada en direcciones IP
- Utiliza la misma configuración para todos los clientes incluyendo clientes Web remotos

Conectividad empresarial MES/ERP (fabricación de sistemas de ejecución/planificación de recursos empresariales)

- Conecta los distintos niveles de la planta a los sistemas de negocios
- Esta construido en estándares SQL, OPC y tecnologías Web
- Build on standard SQL, OPC, and Web technologies for connectivity
- Captura los datos desde una base de Lookout hacia una base datos, hoja de cálculo o software MES

Desempeño en el manejo de eventos

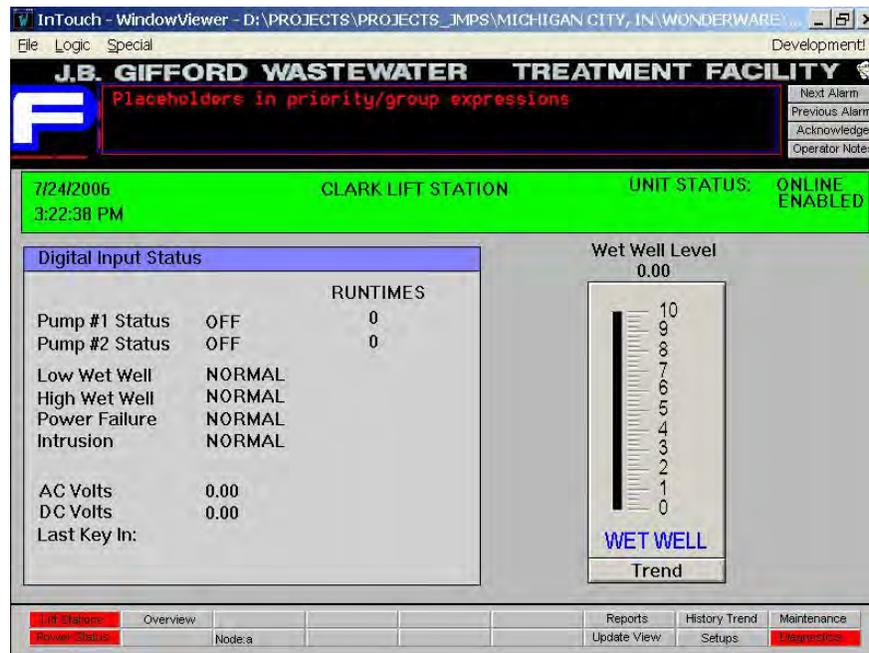
- Reacciona a eventos sustancialmente rápidos
- Interroga a más de 50,000 puntos E/S
- Escala la aplicación con una pequeña perdida en el desempeño.

Visualización

- Utiliza más de 3,300 gráficos pre-construidos
- Habilita desplazamiento en 2D, redimensionamiento y coloreado dinámico

- Importación de gráficos (.bmp) y (.wmf)

Wonderware InTouch



Costo

- 10.000 \$

Costo de mantenimiento anual

- 7,800 \$

Conectividad Abierta

- Wonderware incluye el paquete DA Servers, para crear enlaces entre el servidor de la base de datos y los clientes OPC, este paquete incluye tres tipos de protocolos, DDE, SuiteLink y OPC.
- Contiene los drivers de comunicación para dispositivos tales como Allen Bradley, Mitsubishi, Modicon, Omron, Siemens, etc.

Configuración en línea

- La integración y conectividad de nuevos dispositivos se realiza sin interrupciones.

- La migración a una nueva versión de software no necesita de una nueva configuración.

Alarmas y eventos distribuidos

- Notificación remota de las alarmas para trabajadores aislados geográficamente
- Visualización y reconocimiento de alarmas y eventos desde cualquier parte en la red
- Anuncio local mediante radio difusión de voz para todo el personal y planta
- Información a pedido de la situación del proceso y sus variables

Seguridad

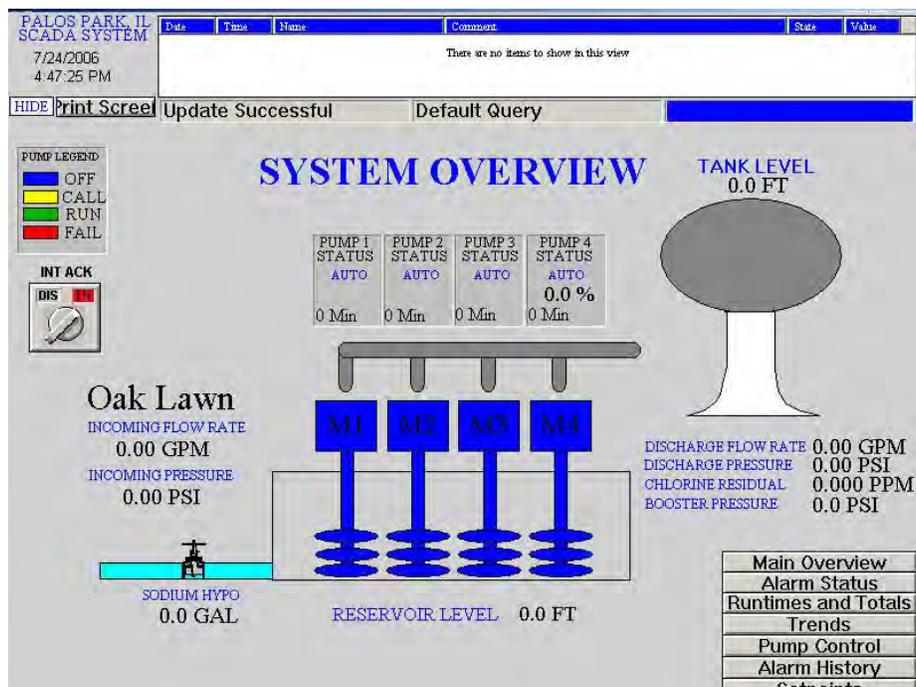
- Bloqueo con password del sistema para que los usuarios autorizados
- Activación de seguridad basada en direcciones IP

Desempeño en el manejo de eventos

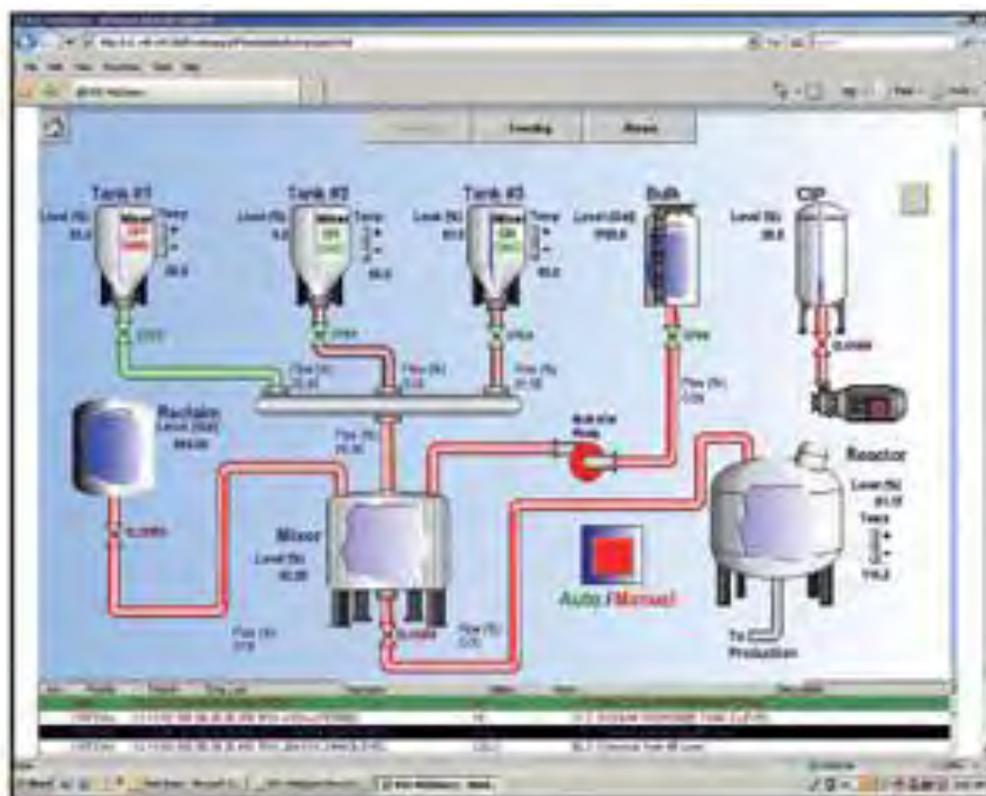
- Tiempo de reacción alto ante eventos con condición de alarma.
- Visualización y análisis de datos históricos en tiempo real.
- 100,000 puntos E/S de enlace a dispositivos, trabajando por eventos.

Visualización

- Utiliza más de 500 gráficos inteligentes y personalizables



Intellution IFIX (General Electric)



Costo

- 200 \$/nodo

Conectividad Abierta

- IFIX ofrece un motor SCADA robusto, con un amplio conjunto de opciones de conectividad, tiene la capacidad de conectarse a casi todo tipo de dispositivo desde su servidor.
- Es completamente independiente del hardware y proporciona flexibilidad en la comunicación con varias marcas desde un único servidor.

Configuración en línea

- IFIX es una solución escalable, con una capacidad de enlazamiento de hasta 200 nodos.
- La integración y conectividad de nuevos dispositivos se realiza sin interrupciones.
- Se puede actualizar de versiones anteriores con la facilidad y funcionalidad que más convenga al usuario.



Seguridad

- Incrementa la seguridad en el rendimiento de las cuentas de usuario, esta seguridad tiene la capacidad de reconocer firmas electrónicas, que restringen el acceso con distintos niveles de seguridad, sistemas de auditoría empresarial.

Visualización

- La nueva versión 5.0 Fluent Ribbon desarrolla una interface de usuario que permite desplegar funciones y soluciones en forma rápida y amigable.
- iFIX acelera drásticamente el proceso de desarrollo de interfaces, aplicaciones con despliegue de ventanas.

Elipse SCADA (Elipse Software)

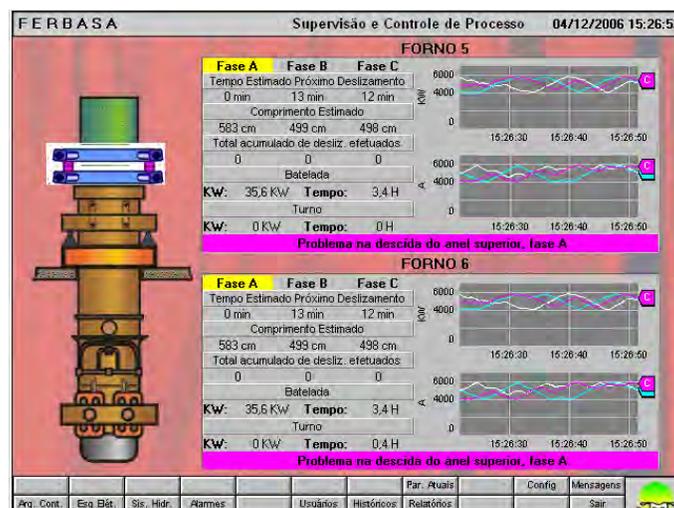


Conectividad Abierta

- Este software está disponible para sistemas que contienen equipos compactos, con poca capacidad de memoria o de memoria solida, tales como equipos portátiles.
- El desarrollo está basado en el sistema operativo Microsoft Windows CE (equipos compactos), tales como celulares, dispositivos PAL u otros dispositivos móviles con capacidad de contener HMIs.

Configuración en línea

- Contiene dos tipos de configuración, la primera es de Desempeño en tiempo real, la cual habilita una aplicación de supervisión previamente desarrollada, para funcionar sin limite de tiempo, esta aplicación no permite cambios.



Capacidad de servicio de plataforma

- Esta disponible para dispositivos basados en las siguientes plataformas: HPC2000 ARM, HPC2000 X86, HPC2000 MIPS y Pocket ARM.

Seguridad

- El sistema de seguridad cuenta con un dispositivo llamado Hardkey, el cual está definido para cada modulo, este es un tipo de clave de acceso el cual puede ser modificado vía e-mail o telefónica.

Desempeño en el manejo de eventos

- Tiempo de entrega de informe basado en el tipo de conectividad con el que cuente el dispositivo.
- Informe de eventos o alarmas mediante generación de históricos, llamadas o mensajes de texto.
- Visualización y análisis de datos históricos en tiempo real.

Paquetes

- **Eclipse View**

Habilita variables de visualización, incluyendo el uso de animaciones, programación de setpoints, control de acceso y funciones especiales de teclado táctil.

Para instalar el sistema se debe contar con los driver DLL del equipo, se puede visualizar alarmas, bloqueo de comunicaciones, intercambio directo de datos, control de acceso a los datos.

- **Eclipse MMI**

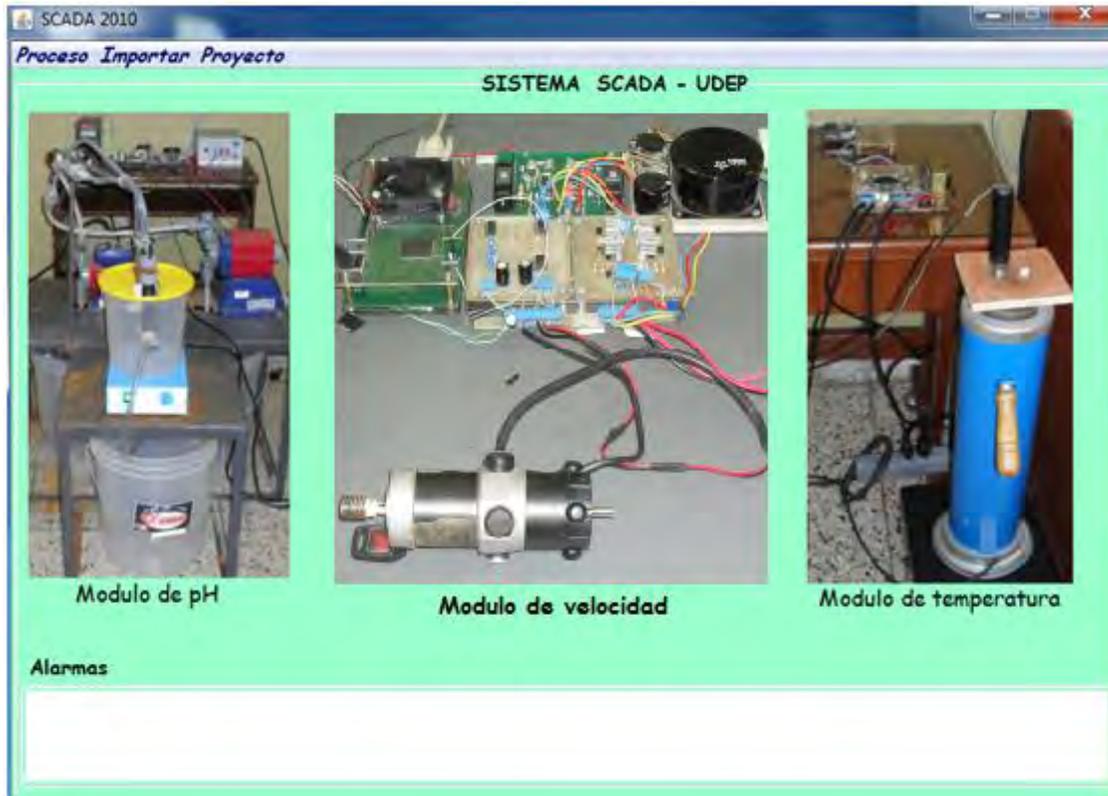
Implementa una base de datos para propietario, es decir envía los datos que están autorizados para ciertos usuarios, los formatos de los reportes, los históricos, las alarmas y el control estático de los procesos, adicionalmente guarda los reportes de alarmas.

Este sistema está orientado para sistemas a medida, donde las conexiones con bases de datos externas (ODBC - conectividad de base de datos abierta) o aplicaciones en la red no son necesarias, y cuando el usuario necesita supervisar otras estaciones.

- **Eclipse PRO**

Esto permite el intercambio de datos en tiempo real con otras estaciones, transferir o descargar bases de datos, desarrollar comandos, programar setpoints vía redes locales o telefónica.

SCADA PRO 2010 (UDEP)



Conectividad Abierta

- Capacidad para cliente y servidor (Ethernet) y conectividad serial.
- Enlace a múltiples objetos en la interface HMI creados en librerías de objetos.
- Conexión al ambiente de programación Java.
- Conexión a dispositivos que contengan el protocolo serial y Ethernet.

Configuración en línea

- Se pueden agregar aplicaciones sin interrumpir la comunicación con los dispositivos.
- Elimina vacíos en los datos, se muestrea por eventos y también por tiempos configurables; se monitorea alarmas.
- Desarrolla y modifica aplicaciones sin entrar a inactividad al proceso.

Alarmas y eventos distribuidos

- Almacena comodato adicional los eventos indeseados tales como alarmas o eventos pre-configurados.
- Visualización y reconocimiento de alarmas y eventos desde cualquier parte en la red

- Utiliza un visualizador de datos común a todos los procesos, en el cual se puede observar en color rojo los eventos anormales de funcionamiento del sistema.

Capacidad multiplataforma

- Disponible para funcionar en Windows XP/7.
- Disponible para funcionar en Linux.

Seguridad

- Bloqueo del sistema para que los usuarios no autorizados no puedan modificar el sistema una vez que entre en funcionamiento.
- Provee niveles de lectura y escritura a procesos, folders y objetos basados en niveles de seguridad.

Desempeño en el manejo de eventos

- Reacciona a eventos sustancialmente rápidos
- Interroga a mas de 20 puntos E/S
- Escala la aplicación con una pequeña perdida en el desempeño.

Visualización

- Utiliza gráficos pre-construidos
- Habilita desplazamiento en 2D, redimensionamiento y coloreado dinámico
- Importación de gráficos