

Universidad de las Ciencias Informáticas
Facultad 5



Título: WebUInt, Interfaz Hombre-Máquina del SCADA en la Web.

Autor: Ariel Guerra Garayta.

Tutor: Ing. Rene López Banacaldo.

29 de Abril del 2009

Resumen

Este trabajo surge a partir de la necesidad que existe en el proyecto “SCADA Nacional” de supervisar y controlar el SCADA desde el nivel de campo de una forma eficiente mediante una aplicación Web. Este trabajo refleja una investigación sobre las tecnologías que más se utilizan en el ámbito de aplicaciones Web, orientándola hacia la automática. El objetivo de este trabajo es obtener una aplicación Web capaz de visualizar y supervisar distintos tipos de gráficos y controles previamente diseñado en el “SCADA Nacional” de una manera eficiente mediante una aplicación confiable y la vez amigable, que permita la interacción con los componentes representados y que sea accesible desde el nivel de campo. Para cumplir esta meta se describe una solución partiendo del diseño de software y se detalla la implementación del sistema.

Índice de contenido

<u>Capítulo 1: Fundamentación Teórica.....</u>	<u>11</u>
<u>1. Introducción.....</u>	<u>11</u>
<u>2. Características de los sistemas SCADA.....</u>	<u>11</u>
<u>3. Elementos de los sistemas SCADA.....</u>	<u>13</u>
<u>4. Funcionalidades de los sistemas SCADA.....</u>	<u>16</u>
<u>5. Módulos de los sistemas SCADA.....</u>	<u>17</u>
<u>6. ¿Qué es un HMI Web?.....</u>	<u>20</u>
<u>7. Ventajas y Desventajas de un HMI Web.....</u>	<u>23</u>
<u>8. Conclusiones.....</u>	<u>25</u>
<u>Capítulo 2: Solución Cliente Web.....</u>	<u>26</u>
<u>1. Introducción.....</u>	<u>26</u>
<u>2. Análisis de los navegadores Web.....</u>	<u>26</u>
<u>3. Componentes del módulo de visualización Web.....</u>	<u>28</u>
<u>4. Solución en el cliente Web.</u>	<u>29</u>
<u>5. Mecanismo de comunicación asíncrona.....</u>	<u>30</u>
<u>6. Estructura de la petición al servidor.....</u>	<u>31</u>
<u>7. Estructura de la respuesta del servidor.....</u>	<u>34</u>
<u>8. Conclusiones.....</u>	<u>37</u>
<u>Capítulo 3: Solución Servidor.....</u>	<u>38</u>
<u>1. Introducción.....</u>	<u>38</u>
<u>2. Servidor Web.....</u>	<u>38</u>
<u>3. Servidor Apache.....</u>	<u>38</u>
<u>4. Construcción de bibliotecas dinámicas en Apache.....</u>	<u>39</u>

5. Módulo de transferencia de datos (DTM).....	40
6. Funcionamiento del DTM.....	41
7. Servidor HMI Web.....	42
8. Patrón de diseño del servidor	42
9. Subsistemas del Servidor HMI Web.....	43
10. Conclusiones.....	45
Conclusiones Generales.....	46
Recomendaciones.....	47
1. Recomendaciones en el cliente Web.....	47
2. Recomendaciones para el módulo de transferencia de datos (DTM).....	47
3. Recomendaciones para el servidor HMI Web.....	47
Referencias bibliográficas.....	49
Anexos.....	50
Glosario de Términos.....	51

Índice de ilustraciones

Representación de un sistema SCADA.	12
Niveles de un sistema SCADA.....	14
Relación entre los componentes del módulo de visualización Web.....	29
Mecanismo de creación de objetos gráficos.....	33
Ventana con objetos gráficos.....	35
Diagrama de Diseño Fábrica Abstracta.....	43

Introducción

La Web y el sistema operativo GNU/Linux se han convertido en el medio de acceso a un gran número de servicios de información. Unos de los campos de esta demanda lo constituyen los sistemas de Supervisión de Control y Adquisición de Datos (SCADA), utilizados para la gestión y el control de procesos industriales. La presencia de estos sistemas aumenta los niveles de eficiencia, minimiza los costos y optimiza los procesos de producción.

El objetivo principal de la automatización industrial es administrar la actividad y la evolución de los procesos sin la intervención continua de un operador humano. En este sentido se han desarrollado los sistemas SCADA, los cuales permiten supervisar y controlar las diferentes variables de un proceso determinado.

El término SCADA usualmente se refiere a un sistema central que monitorea y controla un sitio completo o un sistema que se extiende sobre una gran distancia (kilómetros / millas). La instalación de un sistema SCADA necesita de sensores, actuadores, controladores, redes, comunicaciones, base de datos, Interfaz Hombre-Máquina (*HMI, por sus siglas en inglés*), entre otros. [1]

El sistema permite comunicarse mediante señales de entrada y salida con los dispositivos de campo (sensores, actuadores, controladores) para controlar el proceso desde la pantalla del ordenador, que es configurada por el usuario y puede ser modificada con facilidad. Además, provee de toda la información que se genera en el proceso productivo a diversos usuarios.

La mayoría de los sistemas SCADA, debido a su magnitud y complejidad están formados por diferentes módulos o subsistemas, como son los de: procesamiento de datos, base de datos, manejadores y HMI. En el SCADA que se desarrolla en el polo de Hardware y Automática de la Universidad de la Ciencias Informáticas tiene incorporado el módulo HMI, que está formado por dos sub-módulos, el ambiente de configuración y el ambiente de ejecución, que están en estrecha relación con los demás módulos ya que son los encargados de la interacción del usuario con el sistema.

El ambiente de configuración permite al usuario definir el entorno de trabajo de su aplicación según la disposición de pantallas requerida y los niveles de acceso para los distintos usuarios. Dentro del módulo de configuración el usuario define las pantallas gráficas o de texto que va a utilizar, importándolas desde otra aplicación o generándolas desde el propio SCADA. Para ello, se incorpora un editor gráfico que permite utilizar elementos estándar disponibles (líneas, círculos, textos o figuras). Abarca las utilidades relacionadas con la creación, edición de los elementos que representaran y compondrán el sistema, define la comunicación que utilizara con los dispositivos de campo, mediante los manejadores de comunicación.

El ambiente de ejecución o *run-time* proporciona al operador las funciones de control y supervisión de la planta. El proceso a supervisar se representa mediante sinópticos gráficos generados desde el editor incorporado en el SCADA o importados desde otra aplicación de uso general durante la configuración del paquete. Los sinópticos cambian dinámicamente a diferentes formas y colores, según los valores leídos en la planta o en respuesta a las acciones del operador.

Como ya se había señalado anteriormente, los sistemas SCADA permiten supervisar y controlar diversos aspectos de un proceso determinado; esto avala su importancia y aplicabilidad.

Algunas de las industrias que utilizan los sistemas SCADA son en las de gestión de recursos hidráulicos, energía eléctrica, señales de tráfico, sistemas de control ambiental, control de oleoductos, redes de distribución de gas natural y en sistemas de fabricación, entre otras. Pero esto es sólo a modo ilustrativo, pues existen escenarios muy variados donde se hace necesario la supervisión, el control de procesos y la accesibilidad de forma inmediata a la información de una manera rápida, confiable, segura y con el mínimo coste posible.

Otros beneficios que brindan estos sistemas al personal autorizado de la organización, donde se ponga en marcha el sistema, es el acceso a los comportamientos generales del sistema, como estadísticas del proceso, detección de fugas y la representación de los sinópticos gráficos configurados en los despliegues. Además, que el acceso al sistema pudiera realizarse desde instalaciones distantes de las estaciones de trabajo donde se encuentra instalado el sistema. Donde no se dispone en muchas ocasiones con los medios necesarios para acceder al sistema desde un ordenador central que tenga las

aplicaciones necesarias para visualizar un sistema SCADA. En estos casos se cuenta con dispositivos de baja capacidad de procesamiento, como puede ser: un teléfono móvil con navegación Web, un dispositivo portátil con un sistema operativo embebido; o un ordenador sin ningún software de sistemas SCADA, sin importar el sistema operativo.

En este punto surge la necesidad de encontrar una nueva vía para visualizar los despliegues configurados previamente en la herramienta de configuración del SCADA, de una forma eficiente y segura a los usuarios con los privilegios necesarios, haciendo uso de las ventajas que ofrece las aplicaciones Web, con una funcionalidad total a través de los navegadores estándar, donde se visualizará la información de la planta en una interfaz amigable.

En el Polo de Hardware y Automática (HA) de la Universidad de las Ciencias Informáticas (UCI), se desarrolla el proyecto "SCADA", que tiene como objetivo el control de procesos industriales. Los integrantes de este proyecto tendrán la responsabilidad de crear un sistema SCADA que represente sus interfaces gráficas en la Web, que sea lo suficientemente flexible como para poder integrarlo a partir de la implementación ya existente en el proyecto "SCADA". Añadiéndose a esto que los controles de la Web deben ser semejantes a los del proyecto "SCADA" de escritorio desarrollado en el polo de Hardware y Automática de la Universidad de las Ciencias Informáticas, siguiendo con la estrategia del proyecto de usar paradigmas y plataformas de software libre.

Con estos objetivos y a partir de dicha situación problemática surge el siguiente **Problema** a resolver:

La no existencia de un módulo de visualización Web para el proyecto SCADA que reutilice los objetos gráficos.

Según el Problema expuesto anteriormente se plantea como **objeto de estudio** la visualización de las Interfaces Hombre-Maquinas de sistemas SCADA en la Web.

Para dar solución al problema planteado se ha trazado como **objetivo general**:

Desarrollar un módulo para la visualización de una Interfaz Hombre-Maquina de un sistema SCADA en la Web.

El **Campo de acción** es el desarrollo para la visualización en la Web del proyecto SCADA.

Para cumplir con el objetivo se definieron las siguientes tareas de la Investigación:

1. Revisar bibliografía referente al estado del arte de los sistemas SCADA Web.
2. Analizar las tecnologías existentes para el desarrollo de un HMI en sistemas SCADA Web.
3. Definir las tecnologías a usar.
4. Estudiar los módulos HMI de los sistemas SCADA para definir las funcionalidades a desarrollar.
5. Implementar los elementos que sean necesarios para la estructura de los controles y las vistas de la interacción del usuario con el sistema.
6. Realizar un demostrativo con los controles desarrollados.
7. Valorar los resultados de los demostrativos instalados.

Para el cumplimiento de estos objetivos se llevan a cabo varios **métodos y técnicas en la búsqueda y procesamiento de la información** como son:

Métodos Teóricos.

- **Método analítico-sintético:** Para el estudio de las concepciones y los conceptos empleados en los sistemas SCADA Web, analizando todos los documentos elaborados por desarrolladores, para la extracción de los elementos más importantes.
- **Método histórico-lógico:** Para la comprensión de los antecedentes y las tendencias actuales referidas la evolución en el mundo de los sistemas SCADA Web.

Métodos Empíricos.

- **Experimento:** Elaboración de un HMI en la Web para la visualización de objetos gráficos.

Estructura de los capítulos:

Capítulo 1: En este capítulo se explican las principales características de un sistema SCADA y los módulos por los que esta compuesto, haciendo énfasis en los de visualización. Se describe que es un HMI Web, sus principales ventajas y desventajas.

Capítulo 2: En este capítulo se describen las características, el funcionamiento y el mecanismo de comunicación utilizado en el cliente Web, así como la compatibilidad con los diferentes navegadores Web. Se explica detalladamente el funcionamiento del módulo de visualización Web y los componentes que lo componen.

Capítulo 3: El objetivo de este capítulo es explicar los componentes necesarios del lado del servidor tanto Web como el servidor de HMI Web. Se describe el funcionamiento del módulo de transferencia de datos empujado en el servidor Web Apache, así como sus principales entidades. Además se mencionan las responsabilidades que debe tener el servidor HMI Web y sus principales subsistemas, también se describe el patrón de diseño utilizado para incorporar el nuevo comportamiento de los componentes gráficos para la aplicación Web.

Capítulo 1: Fundamentación Teórica

1. Introducción

En este capítulo se describen los sistemas SCADA y se explican las principales características de estos sistemas en la Web, se realizó un análisis de las tendencias y tecnologías que son más utilizadas actualmente en el desarrollo de aplicaciones Web, así como las ventajas y desventajas de las aplicaciones de supervisión y control en la Web.

2. Características de los sistemas SCADA.

El objetivo principal de la automatización es disminuir la intervención del operador humano en la ejecución de los procesos de producción de las empresas. Con vista a cumplir esa exigente meta se han desarrollado en los últimos años algunos sistemas denominados SCADA [1].

Los sistemas SCADA son aplicaciones de control de producción que se comunican con los dispositivos de campo y controlan el proceso de forma automática desde la pantalla del ordenador, proporcionando de esta forma información del proceso a diversos usuarios entre los que se encuentran los operadores, supervisores, mantenedores entre otros.

Los sistemas de interfaz entre usuario y planta, están basados en paneles de control e indicadores lumínicos, instrumentos de medidas y botones, que están representados en la pantalla de un ordenador.

Provee toda la información que se genera en el proceso productivo a diferentes usuarios: supervisores de control de calidad, mantenedores, asesores, operadores y otros. En este tipo de sistemas interactúan ordenadores, unidades remotas, sistemas de comunicación que efectúan las tareas de supervisión, gestión de los datos y el control total de los procesos.

Otras de las características son la adquisición y el almacenamiento de información en bases de datos de

tiempo real, representación gráfica de los procesos de manera animada, creación de gráficos de tendencia a partir de los valores de las variables en el tiempo, conectividad con otros sistemas (locales y compartidos) y explotación de los datos para la gestión de la calidad, control estadístico, administración y generación de informes.

Son sistemas con arquitectura abierta, capaces de crecer o adaptarse según las características cambiantes de las empresas, que se relacionan con total facilidad y de forma transparente al usuario con los equipos de campo y el resto de los ordenadores involucrados en el proceso, programas en general fáciles de instalar y configurar, con pocas exigencias de hardware y que muestren interfaces amigables a los operadores y demás interesados.

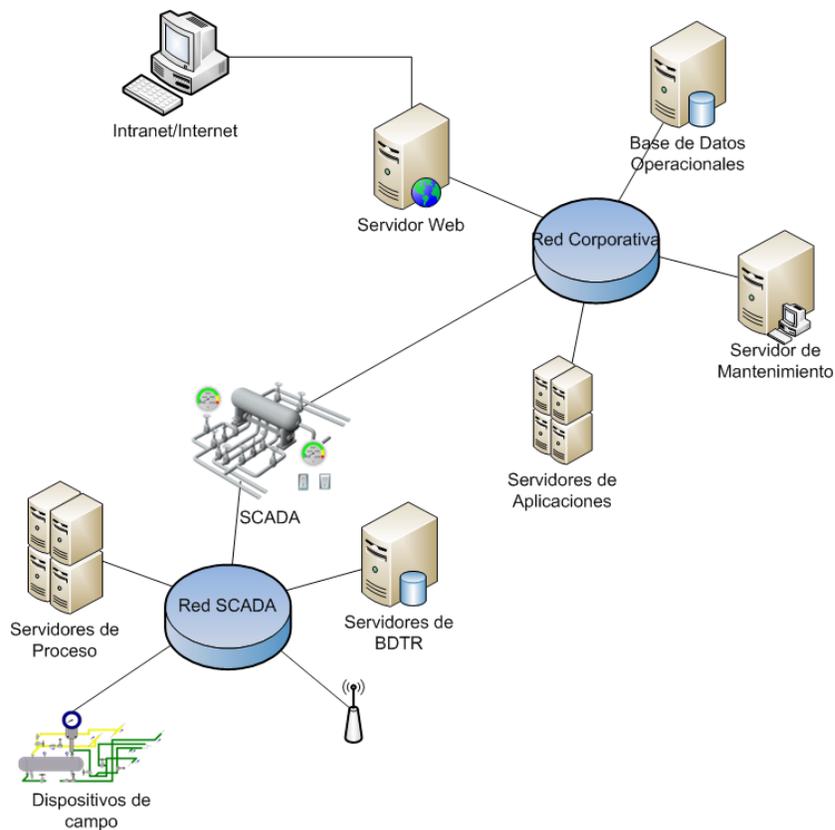


Figura 1: Representación de un sistema SCADA.

3. Elementos de los sistemas SCADA.

Un sistema SCADA está compuesto por 3 elementos fundamentales [1]:

- Múltiples Unidades de Terminal Remota (RTU)
- Estación Maestra y ordenador con interfaz hombre-máquina (HMI)
- Infraestructura de comunicación
- Instrumentación de Campo

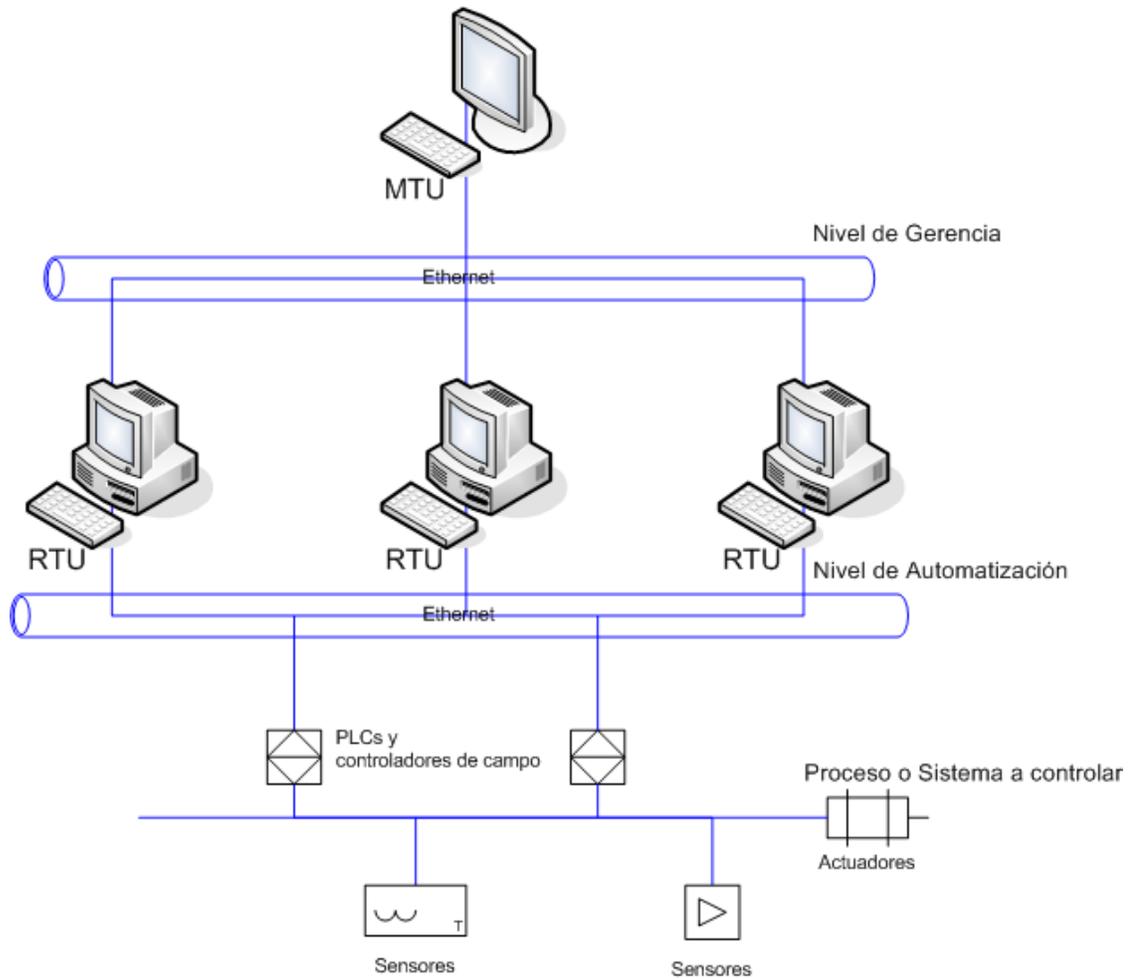


Figura 2: Niveles de un sistema SCADA.

Unidades Remotas (RTU):

Las Unidades Terminales Remotas (RTU), en inglés (Remote Terminal Units), son dispositivos de adquisición de datos y control en campo, cuya función principal es hacer de interfaz entre los equipos de

instrumentación, control local y el SCADA. [1]

Las RTU lee los datos de estado desde los dispositivos de campo (una válvula o un intercambiador) y los interpreta como estados (abierto/cerrado) o medidas ingenieriles (presión, flujo, voltaje o corriente). Las RTU envían señales que pueden controlar a los dispositivos de campo (abrir, cerrar, intercambiar la válvula o configurar la velocidad de la bomba).

Unidad Maestra (MTU)

El Ordenador Central o Unidad Maestra (MTU), en inglés (Master Terminal Unit), es el elemento central de control de un sistema de adquisición de datos y control supervisorio. Este término ha sido comúnmente utilizado para designar al sistema electrónico de computación, que adquiere toda la información procedente de las Unidades Terminales Remotas (RTU) y la presenta para ejecutar una acción de control remoto. La capacidad funcional de una estación maestra incluye todas las tareas de recolección de datos y envío de comandos remotos.

Adicionalmente las funciones de la MTU incluyen el almacenamiento de información histórica, programación, despacho y ejecución de tareas específicas tales como reportes y contabilidad de producción. [1]

Canales o Medios de Comunicación:

La efectividad y confiabilidad operacional de un sistema SCADA depende en gran medida de la transmisión de datos entre la estación maestra y las unidades terminales remotas, por lo tanto, debe ser provisto de un medio a través del cual se establezca el intercambio de datos entre éstas unidades de una forma coordinada, confiable y segura.

El sistema de comunicación que forma parte de un sistema SCADA es el conjunto de elementos, dispositivos y equipos de transmisión de datos a través de los cuales se realiza el intercambio efectivo de mensajes entre las RTUs y la MTU.

Entre los componentes del sistema de comunicación de un SCADA típico, se encuentran las interfaces de comunicación digital, módem, medios de transmisión de datos, el computador frontal de

comunicaciones. (CFE, Communication Front End) y el protocolo de comunicación. [1]

Instrumentación de Campo:

La instrumentación es un área específica de la ingeniería de automatización y control de procesos, la cual comprende la medición de cantidades físicas asociadas a los mismos.

La instrumentación o dispositivos de campo comprenden sensores y actuadores. Un sensor es un dispositivo que detecta, o mide manifestaciones de cualidades o fenómenos físicos, como la energía, velocidad, aceleración, tamaño, cantidad, nivel, flujo, temperatura y presión. Es un dispositivo que transforma la magnitud que se quiere medir, en otra, que facilita obtener una señal eléctrica en un rango manejable por el próximo elemento de la cadena de adquisición de la información. Por otra parte los actuadores son dispositivos que transforman una señal de entrada (eléctrica) en una acción física. [1]

4. Funcionalidades de los sistemas SCADA.

Resulta de interés presentar las principales funcionalidades que se pueden observar en un sistema SCADA.

Un SCADA debe cumplir tres funciones principales:

- *Adquisición* de datos para recoger, procesar y almacenar la información recibida.
- *Supervisión* para observar desde un monitor la evolución de las variables de control.
- *Control* para modificar la evolución del proceso, actuando sobre los reguladores autónomos básicos (consignas, alarmas, menús, etc.) o directamente sobre el proceso mediante las salidas conectadas.

En este tipo de sistemas interactúan ordenadores, unidades remotas, sistemas de comunicación; que

efectúan las tareas de supervisión, gestión de los datos y el control total de los procesos entre otras tareas en las que se encuentran:

- Adquirir información de campo.
- Procesar información de campo.
- Generar alarmas.
- Almacenar la información adquirida por largos periodos de tiempo.
- Generar informes.
- Cambiar estados de los dispositivos de campo.
- Permitir la interacción a través de interfaces gráficas.

Partiendo de lo anteriormente dicho se pueden definir un conjunto de bloques funcionales que aparecen en un SCADA:

- Comunicación con el Campo.
- Procesamiento de la información.
- Almacenamiento de la información.
- Interacción con el Usuario.

5. Módulos de los sistemas SCADA.

Manejadores

Aseguran mediante una interfaz genérica la comunicación del sistema de supervisión y control con los distintos dispositivos que existen en el campo, ya sean autómatas, PLC, sensores inteligentes, etc.

Núcleo de Procesamiento de Datos

Representa el núcleo principal del procesamiento de los datos, es el encargado del procesamiento y análisis de la información recogida del campo a través de los manejadores. Una vez procesada esta información, es enviada al módulo que la requiera.

Base de Datos Históricas

Aquí se implementa el mecanismo encargado del almacenamiento de la información recibida desde el campo, así como la sucesión de alarmas y eventos generados. La información almacenada es utilizada por varias aplicaciones del sistema.

Middleware

El Middleware es la capa de software, que se encarga de la comunicación entre los diferentes procesos distribuidos de medio y alto nivel, que forman parte del sistema SCADA y el principal elemento que caracteriza su complejidad es la gestión de las comunicaciones.

Ambiente de Configuración

Esta aplicación permite configurar varios procesos o partes de ellos, aquí se definen y gestionan las variables, la configuración de los manejadores, los comandos, las alarmas y variadas opciones adicionales. Este ambiente funciona como una aplicación de diseño tradicional, con la peculiaridad que los sinópticos se confeccionan a partir de objetos y primitivas básicas predefinidas, que se pueden agrupar, combinar, transformar, importar y exportar entre otras.

Ambiente de Ejecución (*Run-time*)

Aplicación encargada de la visualización, la cual permite representar gráficamente de manera fidedigna los procesos operacionales con los que el usuario interactúa. A través de esta aplicación se pueden visualizar las condiciones operacionales, valores de variables, navegar entre despliegues, visualizar alarmas, enviar comandos, entre otros.

Interfaz Hombre - Máquina o HMI (Human Machine Interface)

El término Interfaz Hombre-Máquina es usado frecuentemente en el contexto de los Sistemas de Computación y los Sistemas Electrónicos, constituye una capa intermedia que los independiza y permite la intercomunicación entre ambas partes.

Existen en la actualidad diferentes tipos de HMI, se pueden identificar claramente dos de ellas:

Interfaces Gráficas de Usuario (GUI) y las Interfaces de Usuario basadas en la Web. Existen otra variedad de interfaces que actualmente se utilizan en menor escala, por ejemplo: interfaces basadas en líneas de comandos, táctiles, basadas en gestos, multi-pantallas e interfaces basadas en texto.

Los sistemas SCADA de la actualidad poseen varias HMI, estas interfaces emplean técnicas de visualización de elementos gráficos, anteriormente de tipo rasterizadas y de tipo vectorial. Las HMI pueden ser GUI (*Graphical User Interface*) o basadas en Web, dependiendo del interesado en supervisar el proceso en cuestión.

El módulo de HMI en el SCADA se encarga de representar, en un ordenador, los procesos que ocurren en el campo, muestra los componentes implicados, los sensores, las estaciones remotas, y el sistema de comunicación dándole al operador total control. Éste módulo es el que permite al operador estar en contacto directo con el sistema, realizar la supervisión y el control del proceso en general.

Está compuesto por dos partes fundamentales, el ambiente de configuración o editor y el ambiente de

ejecución.

El ambiente de configuración permite configurar varios procesos o partes de ellos, aquí se definen y gestionan las variables, los manejadores, los comandos, las alarmas y variadas opciones adicionales. Este ambiente funciona como una aplicación de diseño tradicional, con la peculiaridad que los sinópticos se confeccionan a partir de objetos y primitivas básicas predefinidas, que se pueden agrupar, combinar, transformar, importar y exportar entre otras.

El ambiente de ejecución se encarga de visualizar las animaciones y los objetos definidos en el editor, muestra lo que está ocurriendo en el campo en tiempo real, es el que envía los comandos a las estaciones remotas, quién recibe los valores de las variables, interactúa con la mayoría de los operadores pues se emplea para supervisar el proceso de manera directa. Al ser el módulo que se encarga de brindar el control total sobre el proceso de producción, la interfaz de usuario brinda un conjunto de funcionalidades primarias, entre ellas la generación de reportes, impresión, análisis de variables, visualización de la tendencia de indicadores, configuración de los manejadores para la comunicación y acceso a las alarmas.

Los HMI representan las actividades que se pueden realizar en un SCADA mediante los despliegues; los despliegues están formados por sinópticos gráficos que toman la forma de los objetos del campo.

Tienen vínculos con bases de datos para proporcionar a los gráficos de tendencias, datos de diagnóstico y manejo de la información así como un cronograma de procedimientos de mantenimiento, información logística, esquemas detallados para un sensor o máquina en particular, incluso sistemas expertos con guía de resolución de problemas.[2]

6. ¿Qué es un HMI Web?

Con el surgimiento y desarrollo de las aplicaciones Web se han abierto las posibilidades en cuanto al acceso y uso de información desde lugares geográficos distantes del mundo. Con los avances en esta tecnología cada vez se demandan aplicaciones más rápidas, ligeras y robustas.

En la actualidad, el acelerado crecimiento de los sistemas de comunicación han hecho de la Internet una

tecnología portadora de una gran variedad de servicios, entre los que se destacan el de correo electrónico, transferencia de ficheros, servicio de información, servicios Web, de televisión y telefonía. También ha permitido que el acceso a estos servicios pueda realizarse desde gran variedad de dispositivos entre los que se encuentran teléfonos móviles, los PDA (*Personal Digital Assistant*), las computadoras, entre otros.

Mediante las aplicaciones Web es posible conocer un evento distante de forma rápida. Facilita el uso de recursos, los servicios y brinda accesibilidad independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura y localización geográfica.

Estos avances han permitido que la Internet sea usada en una amplia variedad de aplicaciones Web y complejos sistemas que antes solo eran posibles con soluciones cliente/servidor de escritorio. La mayoría de los sistemas de supervisión y control de procesos a distancias basados en Internet han enfocado su diseño hacia aplicaciones Web.

El módulo de HMI Web para sistemas SCADA está compuesto un cliente Web y un servidor HMI Web, que se comunican haciendo uso del protocolo de comunicación HTTP (*HyperText Transfer Protocol*). Éste módulo brinda las funcionalidad convencionales de un SCADA, permite a los usuarios autorizados estar en contacto directo con el sistema, realizar la supervisión, la adquisición de datos y el control del proceso en general.

Cliente Web

El cliente Web es una aplicación Web que funciona sobre un navegador Web estándar, recibe y muestra los datos del servidor HMI Web. Mediante la aplicación cliente los usuarios interactúan con el proceso. Una característica especial del cliente Web de HMI es que es un cliente ligero, que no requiere ningún software adicional para realizar las funciones convencionales del SCADA.

Permite acceder a la información proporcionada por el servidor HMI Web. La comunicación con el servidor HMI Web se realiza a través del protocolo HTTP, el cual permite la transferencia de archivos hipertexto.

Protocolo de comunicación HTTP

El Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol*) es un protocolo cliente-servidor que manipula los intercambios de información entre los clientes Web y los servidores HTTP.

Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Una vez que se establece la conexión, el protocolo TCP (*Transmission Control Protocol*) se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

Una transacción HTTP está formada por un encabezado seguido, opcionalmente, por el dato. El encabezado especificará la acción requerida del servidor, el tipo de dato retornado, o el código de estado.

El uso de campos de encabezados enviados en las transacciones HTTP le da gran flexibilidad al protocolo. Estos campos permiten que se envíe información descriptiva en la transacción, permitiendo así la autenticación, cifrado e identificación de usuario.

Un encabezado es un bloque de datos que precede a la información de la transacción, por lo que muchas veces se hace referencia a él como meta dato.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones Web necesita frecuentemente mantener estado. Para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones Web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

En general, una aplicación que inicia una comunicación con otra se la califica como cliente. Los usuarios finales invocan aplicaciones cliente cuando utilizan un servicio de red. Cada vez que se ejecuta una aplicación cliente, esta contacta con el servidor, le envía una solicitud de servicio y espera la respuesta o resultados del servicio. El proceso cliente es el encargado de llevar a cabo la interacción con el usuario y de mostrar los resultados de las peticiones de servicio. En la mayoría de las ocasiones los clientes son mas fáciles de diseñar que los servidores, y no suelen precisar privilegios especiales del sistema para

poder funcionar.

Servidor Web

Un servidor es un programa que espera peticiones de servicio por parte de un cliente. Recibe la petición del cliente, procesa la información, ejecuta el servicio solicitado y retorna los resultados al cliente. No existe una interacción directa entre el usuario y el servidor, de esto ya se encarga la aplicación cliente.

Los servidores HMI Web gestionan la información que le llega desde los dispositivos de campo, el manejo de eventos y crean las representaciones de los sinópticos que luego se visualizara en el cliente Web. El servidor es el encargado de crear, modificar y eliminar los componentes gráficos de la aplicación, atendiendo a las peticiones de los usuarios y las actualizaciones de los valores de las variables de los dispositivos del campo.

Otra funcionalidad de los servidores HMI Web es que provee los datos del campo y los valores que se almacenan en base de datos histórica de forma inmediata. Gestionan el sistema de comunicación dándole al operador control total mediante un navegador Web.

7. Ventajas y Desventajas de un HMI Web.

Ventajas:

- No requieren instalación, pues usan tecnología Web, lo cual nos permite el aprovechamiento de todas las características del Internet.
- Son fáciles de usar, los usuarios con conocimientos básicos sobre el sistema a operar puede manipular la aplicación.
- Alta disponibilidad, ya que puede realizar consultas en desde cualquier regionalización geográfica donde tenga acceso a Internet y a cualquier hora.

- Datos centralizados y fácil integración de datos de múltiples fuentes.
- Se obtiene una reducción de costos, puesto que se racionaliza el trabajo, se reduce el tiempo y dinero dedicado al mantenimiento.
- Menos requerimientos de memoria: Las demandas de memoria RAM (*Random Access Memory*) por parte del usuario final son más razonables que en los programas instalados localmente.

Desventajas:

- Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera excesivo hasta que se obtiene la reacción del sistema.
- Elementos de interacción muy limitados. En comparación con el software de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas, aunque se ha avanzado mucho con la introducción del AJAX (*Asynchronous JavaScript And XML*) en la construcción de páginas Web.
- Diferencias de presentación entre plataformas y navegadores. La falta de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

8. Conclusiones

En este capítulo se abordaron conceptos importantes para entender el funcionamiento de un sistema SCADA. Se explico como está formado un modulo de visualización Web para aplicaciones de supervisión. Se mencionaron las principales ventajas que reporta un módulo de visualización para los servicios Web y sus desventajas para aplicaciones de este tipo. Se describió el tipo de comunicación que deben tener estas aplicaciones con los módulos de los sistemas SCADA.

Capítulo 2: Solución Cliente Web

1. Introducción

En el capítulo anterior se plantearon algunas características que deben tener los componentes de una aplicación de supervisión HMI Web para sistemas SCADA. En este capítulo abordaremos sobre la compatibilidad de los navegadores con las aplicaciones Web. El comportamiento del módulo de visualización Web, el funcionamiento de la aplicación cliente Web y las funcionalidades que debe cumplir.

2. Análisis de los navegadores Web

Un navegador Web es una aplicación software que permite al usuario recuperar, visualizar documentos e interpretar documentos escritos en el lenguaje HTML (*HyperText Markup Language*). Los navegadores actuales permiten mostrar o visualizar la información contenidas en las páginas Web en forma de texto, imágenes, audio, secuencias de vídeo, animaciones, entre otras. Adicionalmente permite descargar documentos seleccionados, capturar y enviar la información solicitada al usuario.

Internamente los navegadores están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS (*Cascading Style Sheets*) para mostrar las páginas se denomina motor de renderizado.

Otro componente de gran importancia es el intérprete de JavaScript, el cual interpreta y ejecuta el código fuente del lenguaje JavaScript, acorde a las instrucciones dadas. Mediante este componente se podrá acceder y modificar el DOM (*Document Object Model*) de un documento HTML o XML (*Extensible Markup Language*). La versión de un motor es mucho más importante que la versión del propio navegador.

A continuación unas tablas comparativas entre los diferentes navegadores en cuanto a la versión de CSS y JavaScript que soportan:

Comparación entre los Navegadores en cuanto a CSS.

Navegador	Internet Explorer	Firefox	Safari	Opera	Chrome
Motor	Trident	Gecko	WebKit	Presto	WebKit
CSS 1	6.0	Si	Si	Si	Si
CSS 2.1	7.0	Si	Si	Si	Si
CSS 3	No	Parcial	Parcial	Parcial	Parcial

Tabla 1: Características de los intérpretes CSS de los navegadores Web.

Comparación entre los Navegadores en cuanto al Lenguaje JavaScript.

Motor render	Trident	Gecko	WebKit	Presto
Motor de JS	JScript	SpiderMonkey	JavaScriptCore	Futhark
Edition 3	Si	0.6	Si	7.0
JavaScript 1.5	No	0.6	Si	7.0
JavaScript 1.6	No	1.8	Parcial	No
JavaScript 1.7	No	1.8.1	No	No
JavaScript 1.8	No	1.9	No	No

Tabla 2: Características de los intérpretes JavaScript de los navegadores Web.

Como se pudo observar en las tablas comparativas existe una gran variedad de intérpretes del lenguaje JavaScript y motores de renderizado gráfico, lo cual dificulta la visualización y la ejecución de script de una forma estándar en todos los navegadores Web. De forma general los intérpretes HTML de los navegadores han alcanzado un nivel similar entre ellos, aunque en la actualidad algunos han añadido funcionalidades de la versión 5 de este lenguaje.

Otra característica de gran importancia en los navegadores Web es el tiempo de espera abriendo una

página Web, así como el consumo de memoria del navegador. Todos estos son factores importantes a la hora de elegir un navegador Web como software cliente de una aplicación Web.

Entre los navegadores mas completo tenemos a Firefox, que cumple con la mayoría de los estándares de la W3C (*World Wide Web Consortium*). Su motor de renderizado Gecko soporta funcionalidades de CSS 3 y la representación de imágenes SVG, PNG, entre otras. Su intérprete de JavaScript SpiderMonkey en su última versión ha incorporado funcionalidades de la versión de JavaScript 1.8.

Por todas las característica antes vistas en los navegadores Web, la aplicación Web debe tener una implementación que tenga el mismo comportamiento en la mayoría de los navegadores para garantizar una portabilidad del software hacia cualquier navegador. Para mas información de los navegadores Web ver tablas comparativas en los anexos.

3. Componentes del módulo de visualización Web

El módulo visualización Web esta formado por diferentes componentes que contribuyen a la comunicación y el manejo de datos con el resto de los módulos del SCADA (manejadores, bases de datos, procesamiento de datos), a continuación se mencionan y se describe la relación que existe entre ellos.

El componente Servidor HMI Web es el módulo que se encuentra en el sistema SCADA que maneja el pintado de los sinópticos gráficos, los comandos, la actualización de los objetos gráficos y la interacción con la capa de comunicación del SCADA que interactúa con el resto de los módulos del SCADA.

Para la transferencia de datos entre el servidor Web y el servidor HMI Web se desarrollo un módulo de transferencia de datos DTM (*Data Transfers Module*), que enviara las peticiones del cliente HMI Web al Servidor HMI Web y viceversa.

Un cliente HMI Web es una aplicación Web que funciona sobre un navegador Web estándar. Que debe mantener una comunicación con el servidor de HMI Web. Para el funcionamiento de la aplicación cliente

HMI Web, se desarrollaron un conjunto de funcionalidades acorde con el soporte de los navegadores Web estándar.

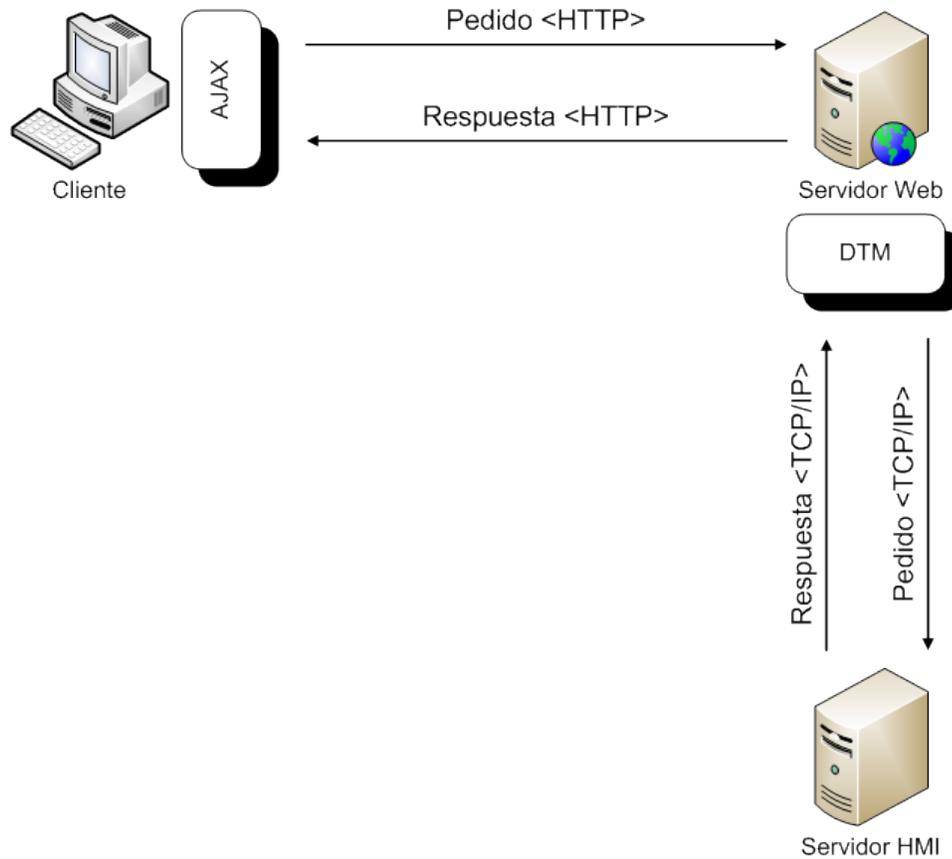


Figura 3: Relación entre los componentes del módulo de visualización Web.

4. Solución en el cliente Web.

Entre las características que debe cumplir una aplicación Web de supervisión están las de brindar una comunicación continua con el servidor de HMI Web, la visualización de los sinópticos gráficos y la

interacción con ellos, permitir la navegación entre los distintos despliegues, la ejecución de mandos y la actualización de los objetos gráficos. Mediante estas características la aplicación logra una comunicación transparente entre el proceso y el usuario con conocimientos básicos de supervisión.

El módulo de visualización Web cuenta de tres componentes principales para cumplir con las características anteriormente dichas, estos componentes son: Cliente Web, Módulo de Transferencia de Datos y el Servidor HMI Web.

5. Mecanismo de comunicación asíncrona

Las aplicaciones Web siguen el paradigma cliente/servidor, en la que el cliente Web inicia la comunicación con el servidor Web, le envía una solicitud de servicio y espera la respuesta o resultados del servicio. El proceso cliente es el encargado de llevar a cabo la interacción con el usuario y de mostrar los resultados de las peticiones de servicio. El servidor recibe la petición del cliente, ejecuta el servicio solicitado y retorna los resultados al cliente.

No existe una interacción directa entre el usuario y el servidor, de esto ya se encarga la aplicación cliente. Por lo que si el servidor tiene un cambio de estado en alguno de los objetos que maneja, hasta que el cliente no pida este cambio el servidor no podrá enviarlo.

Esto trae como consecuencia que si hay un cambio de estado en alguna de las variables de los dispositivos de campo, como puede ser la temperatura, la presión o el consumo energético, el cambio de estado se le notifica al servidor de HMI Web de inmediato, pero el servidor no va a poder notificar dicho cambio hasta que el cliente no lo solicite.

Para solucionar este inconveniente se pueden usar las tecnologías de AJAX (*Asynchronous JavaScript And XML*). Es una funcionalidad que se ejecuta en el navegador de los usuarios y mantiene la comunicación asíncrona con el servidor en un segundo plano. De esta forma es posible actualizar las páginas Web, sin necesidad de que el usuario pida hacer un cambio y mantener a la aplicación cliente informada de los cambios de estado en el servidor. Lo que significa que aumenta la interactividad, velocidad y usabilidad en las aplicaciones.

AJAX es una técnica válida para múltiples plataformas y utilizable en varios navegadores Web dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Funcionamiento del cliente Web.

El operador al realizar la petición de visualización de un despliegue a la aplicación Web, la aplicación realiza un pedido (*Request*), a través de la URL, de dicho despliegue al servidor de HMI. Este pedido va a estar conformado por el id del objeto que es un identificador único para cada elemento del sistema, el tipo de evento que se realizó y el ultimo parámetro es un mecanismo de actualización para la página Web, que realiza AJAX.

6. Estructura de la petición al servidor

A continuación se explican los elementos que conforman la cadena de petición al servidor y como se realiza el pedido.

“?**action**=[evento]&**objid**=[identificador]&**var**=[aleatorio]”

El primer parámetro va hacer la acción (*action*) que se realiza en la aplicación Web, donde recibe como valor el tipo de evento que se va ha realizar (pintado, navegación, actualización). El segundo parámetro es el identificador del objeto que realiza o sobre el que se realiza el pedido al servidor. El último parámetro es un mecanismo de actualización utilizado en AJAX, que mas adelante se explicara.

Tipos de eventos.

Los eventos pueden ser de pintado, de navegación y los de actualización, estos eventos te permiten diferenciar entre los tipos de acciones que se van a realizar en la página Web.

El evento de pintado se utiliza cuando el cliente realiza la petición de construcción de un objeto gráfico,

en este caso la acción que se realiza es la número 1.

El evento de navegación permite al usuario notificarle al sistema que va a cambiar de pantalla, la acción que se realiza en este caso es la número 4.

El evento de actualización se llama cada vez que un objeto o grupo de ellos necesite saber si hubo un cambio de estado en el servidor o el usuario ordene su actualización, la acción que se realiza para este evento sería la número 3.

Los objetos van a poder ser diferenciado a través de los identificadores únicos que se les definen a cada objeto en la aplicación. El mecanismo actualización se utiliza para asegurar que cada pedido que se realice sea distinto al anterior y no se devuelva por defecto la página Web que quedó almacenada en la cache del sistema.

Para realizar una petición de pintado de un objeto:

```
1 var myUrl = "?action=1&objid=" + element.id + "&var=";  
2 myUrl += parseInt( Math.random()*99999999 );  
3 loadContent( myUrl, "GET", showElements );
```

Para realizar una petición de actualización:

```
1 var myUrl = "?action=2&objid=" + element.id + "&var=";  
2 myUrl += parseInt( Math.random()*99999999 );  
3 loadContent( myUrl, "GET", showScreen );
```

Para realizar una petición de navegación de mediante el menú:

```
1 var myUrl = "?action=4&objid=" + element.id + "&var=";  
2 myUrl += parseInt( Math.random()*99999999 );  
3 loadContent( myUrl, "GET", showElements );
```

En los fragmentos de código anteriores del cliente Web se define el tipo de evento que se va a realizar y seguido se el identificador del elemento que realiza el evento. Luego se crea el mecanismo de

actualización para el refrescado de la cache de la aplicación Web. Finalmente se llama a la función encargada de enviar la petición, recibir la respuesta y mostrarla. Esta función recibe como parámetros la URL (*Uniform Resource Locator*) que sería la cadena de petición, el segundo valor sería la forma de enviar los datos (GET o POST) y el último valor sería la función encargada de visualizar la respuesta.

En los ejemplos que se muestran anteriormente con los diferentes valores del **action** que se van a realizar sobre un objeto, se puede lograr varias de las funcionalidades requeridas para el sistema. Las imágenes que se van a visualizar el evento que van a solicitar a través de su URL va hacer de pintado. Para mantener actualizado al sistema en general y a cada elemento que lo compone, la aplicación deberá realizar un pedido de actualización del objeto que se desea actualizar, a través de AJAX. A través de estos eventos el usuario podrá realizar un pedido de navegación hacia otra pantalla, lo cual permitirá al operador tener una visión de todo el proceso que se este mostrando. El método **loadContent** permite definir distintos tipos de función para visualizar la respuesta del servidor de los diferentes tipos de pedidos realizados.

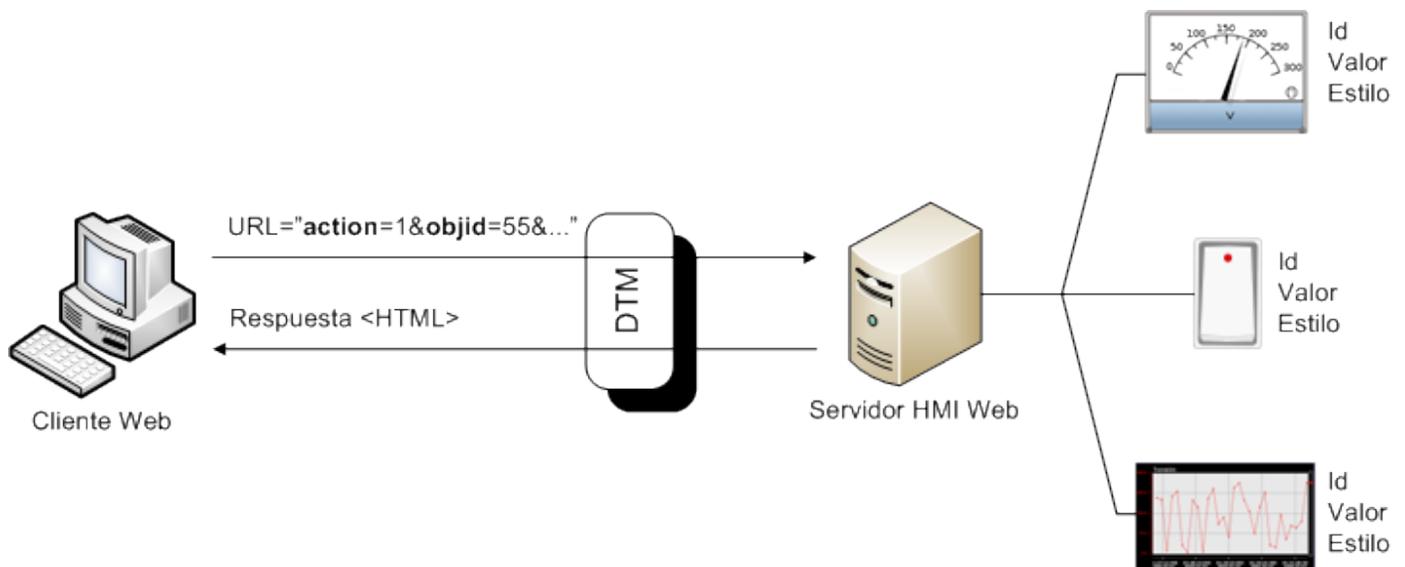


Figura 4: Mecanismo de creación de objetos gráficos.

Las peticiones van hacer procesada y enviada a la capa de comunicación (*Middleware*) del SCADA a través del servidor HMI Web. El servidor confecciona la respuesta (*Response*), a partir de la información recibida de los demás módulos del sistema mediante el *Middleware*. La respuesta va a estar conformada por el código HTML correspondiente al objeto del pedido realizado. El cliente recibe la respuesta del servidor y visualiza los elementos recibidos.

Mediante estos métodos es posible realizar la interacción con el sistema, captura de eventos, ejecución de mandos, actualización, representar objetos contenedores y visuales que usara el sistema para la visualización del proceso. Dentro los componentes contenedores tenemos: ventanas, pantallas, barras de herramientas y barra de menú, entre los componentes visuales se encuentran: gráficas de tendencias, caja de texto, notas, imágenes, botones, medidores verticales y horizontales, menús, entre otros.

7. Estructura de la repuesta del servidor

Cada componente tendrá una estructura en HTML haciendo uso de las etiquetas estándares y un formato de estilo usando CSS, quedando conformada la página en forma de árbol y una correcta apariencia visual. Esto permitirá la portabilidad de nuestro sistema hacia diversos navegadores y una mejor integración con el diseño con la aplicación de escritorio. Con este mecanismo se podrá representar propiedades de los componentes definidos en la aplicación de escritorio.

En los siguientes fragmentos de códigos se muestra como queda conformada la estructura de una ventana y sus propiedades visuales.

```
<HTML>
<HEAD>
...
</HEAD>
<BODY>
```

```

<div id="W34" name="screen">
  <div id="w39">
    <IMG src=" ?action=1&objid=w39&var=58909" style=" " >
  </div>
</div>
</BODY>
</HTML>

```

En este ejemplo se hace uso de los identificadores únicos para cada objeto que se representa, tanto contenedores como objetos gráficos. Donde el objeto contenedor de la pantalla (*screen*) que se mostrara contiene varios elementos gráficos, que mediante los estilos de CSS van a tener una posición en la pantalla, profundidad, fuente, color de fondo, entre otras propiedades visuales que brinda este lenguaje.

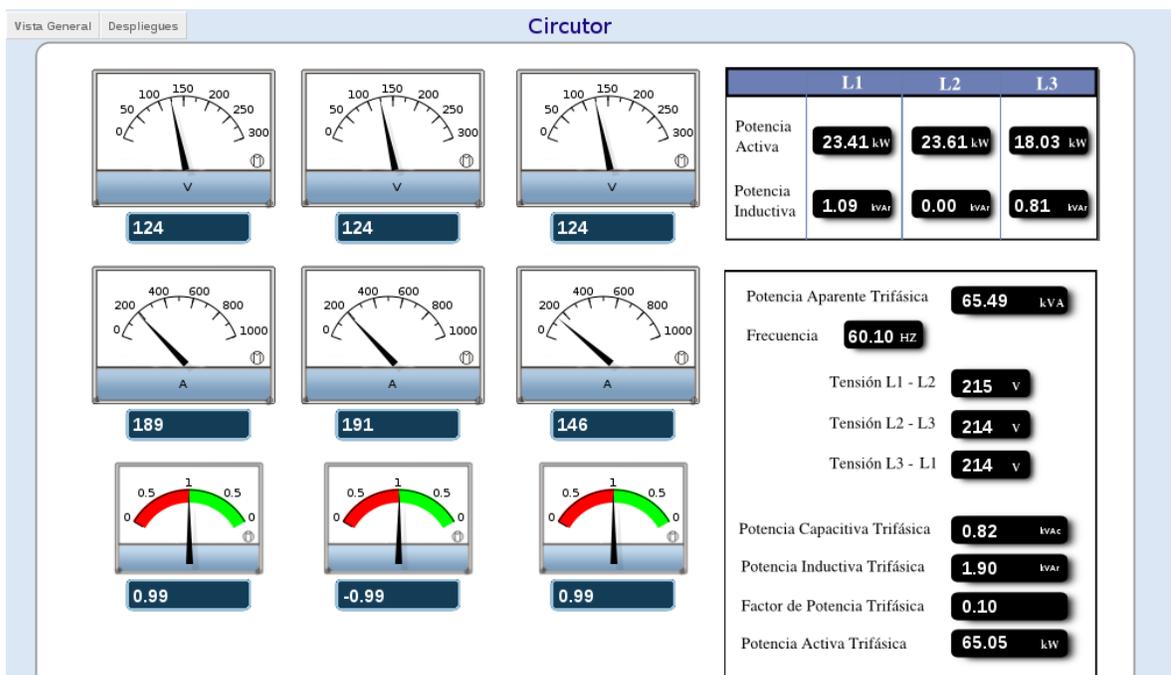


Figura 5: Ventana con objetos gráficos.

En la imagen que se muestra anteriormente se puede observar una ventana con varios objetos gráficos, distribuidos por diferentes áreas de la pantalla. Para ver más imágenes de la aplicación Cliente Web con los diferentes sinópticos gráficos ir a los anexos.

8. Conclusiones.

En el presente capítulo se selecciono a Firefox como uno de los navegadores más completo, ya que cumple con la mayoría de los estándares de la W3C y sus motores de renderizado y de interprete JavaScript son unos de los mas eficientes. Se describieron las principales características del componente Cliente Web y se mencionaron los principales componentes por el que esta compuesto el módulo de visualización Web (Cliente Web, Módulo de Transferencia de Datos y el Servidor HMI Web).

Capítulo 3: Solución Servidor

1. Introducción

En el presente capítulo se abordarán las características y el funcionamiento de los componentes del servidor HMI Web. También se describe el mecanismo de comunicación entre el servidor Web y el servidor HMI Web del sistema SCADA, así como el desarrollo del módulo de transferencia de datos DTM (*Data Transfers Module*) para el servidor Web Apache.

2. Servidor Web.

Un servidor Web es un programa que implementa el protocolo HTTP (*Hypertext Transfer Protocol*). Este protocolo pertenece a la capa de aplicación del modelo OSI (*Open System Interconnection*) y está diseñado para transferir hipertexto, páginas HTML (*Hypertext Markup Language*).

Un servidor Web se mantiene a la espera de peticiones HTTP por parte de un cliente HTTP o navegador Web. El cliente realiza una petición al servidor, la procesa y responde con el contenido que el cliente solicita.

3. Servidor Apache.

Apache es un servidor de páginas Web de código abierto multiplataforma y modular, se desarrolla dentro del proyecto HTTP Server (httpd) de la empresa Apache Software Foundation. Se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios Web elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. [5]

Características que lo favorecen:

- Es una tecnología de código abierto.
- Trabaja sobre múltiples plataformas (Unix, Linux, MacOSX, Vms, Win32, OS2, etc.)
- Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con las API de programación de módulos. Permite el desarrollo de módulos específicos en C o Perl.
- Permite personalizar la respuesta ante posibles errores. Para que ejecute un determinado script cuando ocurra un error en concreto.
- Creación y gestión de mensajes de error altamente configurables, facilitando de este modo el control de las acciones realizadas en el servidor.

4. Construcción de bibliotecas dinámicas en Apache

Una de las principales características de Apache, es su extensibilidad. Esto es gracias a su arquitectura modular, que permite añadirle funcionalidades al servidor mediante módulos. Existen módulos para Apache, oficiales y no oficiales. Algunos se compilan por defecto al compilar Apache. Otros tienen que instalarse explícitamente, ya sea al compilar Apache o dinámicamente mediante bibliotecas dinámicas y la herramienta de extensión de Apache *apxs* (*Apache extension tool*).

apxs (*Apache extension tool*)

Es una herramienta para construir e instalar módulos de extensión para Apache. Permite construir módulos DSO (*Dynamic Shared Object*) de modo que puede añadirse un nuevo módulo sin tener que recompilar Apache, mediante la adición del mismo con la directiva "*LoadModule*". Para poder utilizar esta utilidad, tenemos que tener soporte DSO en Apache, o lo que es lo mismo, tener cargado el módulo "mod_so". [6]

5. Módulo de transferencia de datos (DTM)

El DTM es una biblioteca dinámica que se desarrolló para Apache, usando las API (*Application Programming Interface*) que proporciona el servidor y la herramienta de extensión del Apache. Se encarga de la transferencia de datos entre el servidor Apache y el componente HMI Base del SCADA.

En este módulo se han definido un grupo de entidades de suma importancia en el sistema, ya que proveen los mecanismos para actualizar la información de la aplicación. Entre los elementos más significativo del módulo se encuentra el manejador para las API de Apache y la entidad *tcp_conect* para la comunicación, encargada de la de comunicación a través del protocolo TCP/IP con el servidor HMI.

Entidad de comunicación TCP/IP.

Estas funcionalidades permiten la comunicación a través del protocolo TCP/IP con el servidor de HMI, mediante un grupo de funciones entre las que se encuentran: *tcp_receive_frame*, *tcp_send_frame*, *tcp_connect*, *tcp_close*.

tcp_receive_frame: Función utilizada para el recibo de tramas de información por el puerto de comunicación del servidor.

tcp_send_frame: Función utilizada para el envío de información hacia la dirección IP (*Internet Protocol*) del servidor HMI y del puerto de interfaz genérica del servidor, a través del protocolo de comunicación TCP/IP.

tcp_connect: Función necesaria para iniciar la comunicación con el servidor.

tcp_close: Función necesaria para terminar la comunicación TCP/IP con el servidor.

Manejador de las API de Apache (*hmi_web*)

Apache, en su API, maneja diversas estructuras para gestionar la información según el ciclo de una

petición. Mediante las funciones manejadoras (*handlers*), Apache interactúa con el módulo, en el momento que se realice la petición por parte de los usuarios.

6. Funcionamiento del DTM

El módulo captura las peticiones de la URL que llegan al Apache, inicia la comunicación TCP/IP a través de la función *tcp_connect*, con el componente HMI Web del SCADA, mediante la dirección IP (*Internet Protocol*) y el puerto (3000) previamente configurado. Luego de establecer la comunicación, envía utilizando el método *tcp_send_frame*, la petición realizada del cliente Web por el método GET o POST al servidor Web. En este punto el modulo queda a la espera de la respuesta del servidor HMI Web, por el mismo puerto y dirección IP se realizo el envío, haciendo uso de la función *tcp_receive_frame*.

En el servidor HMI Web mediante la clase *Protocol* utilizando la función *tcp_receive*, del protocolo TCP/IP, se recibe la información, del pedido, que le envía el DTM al puerto previamente seleccionado (3000). Para el envío de información del SCADA al DTM se utiliza la función *tcp_send* del protocolo TCP/IP, que mediante el puerto 3000 le hace el envío de la respuesta de la petición del módulo DTM.

A continuación se muestran fragmentos de código del componente *hmi_web* que se encarga de la comunicación en el DTM, donde se muestra el envío y el recibo de información a través del protocolo TCP/IP.

```
int socket_id = tcp_connect( hmi_web.ip, hmi_web.port );
if ( socket_id < 0 )
{
    ap_rputs( "failed on establishing a tcp connection\n", r );
    return 1;
}
```

```
}  
  
tcp_send_frame( socket_id, by_get_ptr, by_get_length, by_post_ptr, by_post_length, 1000 );  
  
tcp_receive_frame( r, socket_id, hmi_web.port );  
  
tcp_close( socket_id );  
  
return OK;
```

7. Servidor HMI Web

El servidor HMI Web es el encargado de interactuar con el resto de los módulos del sistema SCADA mediante la capa de comunicación. El servidor será el encargado manejar las peticiones y los eventos enviados desde el cliente a través del módulo desarrollado para el Apache DTM. Es el encargado realizar la respuesta a las peticiones del cliente. Permite crear, actualizar y modificar los sinópticos que se visualizaran en el cliente, utilizando los subsistemas creados en el HMI del SCADA (Graphics, Base, Events, Web, etc.), haciendo uso del patrón de diseño Fábrica Abstracta.

8. Patrón de diseño del servidor

Fábrica Abstracta (*Abstract Factory*) es un patrón de diseño para el desarrollo de software. El problema que intenta solucionar este patrón es el de crear diferentes familias de objetos. Uno de los usos más comunes es el de creación de interfaces gráficas de distinto tipo (gtk, qt, web, etc.). Proporciona una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas.

Mediante este patrón podemos adicionar una nueva fábrica de objetos, para el servicio Web, reutilizando las características de los objetos base del sistema. De esta forma los objetos gráficos de las distintas

fábricas van a tener un comportamiento similar.

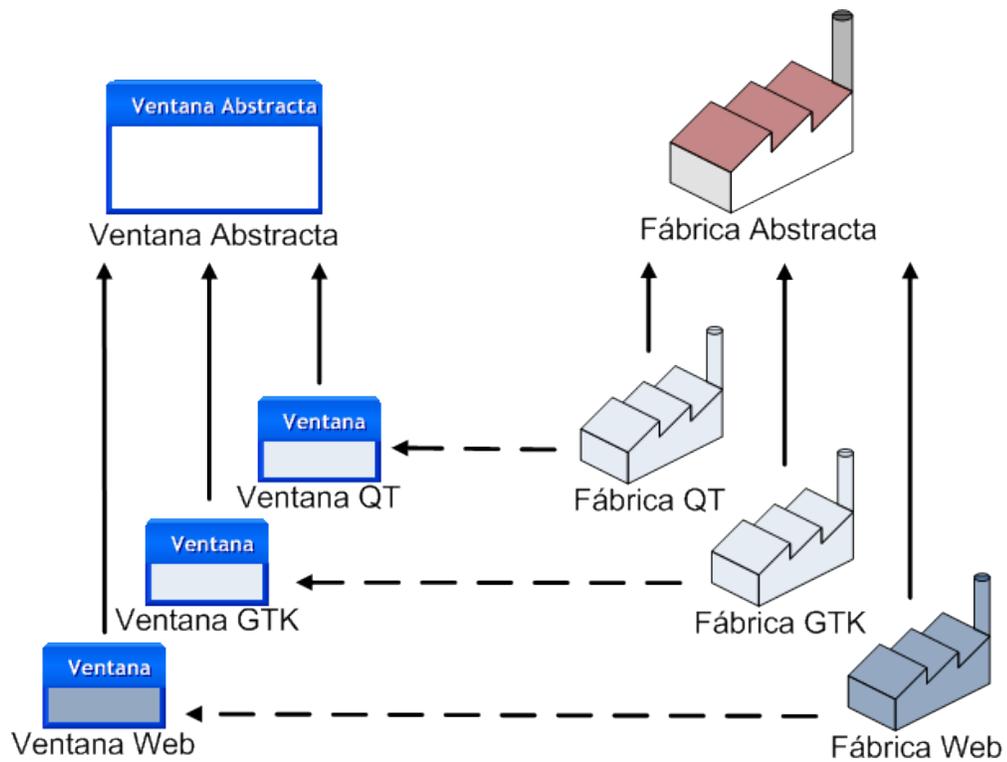


Figura 6: Diagrama de Diseño Fábrica Abstracta

9. Subsistemas del Servidor HMI Web

Subsistema Base

En este subsistema se encuentran las clases sobre las cuales se soportara parte del sistema, estas definen las principales interfaces que se implementarán para la visualización de los despliegues, asignación a las ventanas, fabricación y renderizado de los objetos y el control de la aplicación en general.

Como principales elementos de este subsistema se encuentran las clases Factory, Viewer, Application, Window y Screen, encargadas de crear los objetos, administrar la entrada/salida de los datos, controlar la aplicación, manejar el concepto de ventana y el de pantalla o despliegue respectivamente.

Subsistema Web

En este subsistema se encuentran las clases con las que se construyen los objetos que se representaran en el cliente Web (mayormente imágenes como botones, termómetros, tendencias y otros dispositivos), también se encuentra la implementación de los objetos contenedores (ventanas, pantallas, barras de menú y capas). Se definen funcionalidades importantes de eventos para la Web heredada del subsistema Base.

Mediante la creación de una fábrica de objetos para la visualización en la Web, que estén relacionados con los demás objetos gráficos, se puede lograr un comportamiento similar en los distintos objetos gráficos. Entre los objetos gráficos desarrollados se encuentran los gráficos de tendencias, botones, imágenes, medidores verticales, horizontales, circulares, menús, entre otros.

Para crear una aplicación en el servidor es necesario especificar el puerto de comunicación con el DTM para la sincronización de los componentes. Al desarrollar cada componente del sistema es necesario transformar las imágenes creadas en el servidor en cadenas de bit, para transmitirla hacia el DTM.

Otro elemento importante es la creación de un nuevo evento en la tabla de eventos del sistema (*WebAction*). Se encarga de estar a la escucha de los eventos que llagan desde la Web para los objetos Web. De esta manera se pueden manejar las acciones realizadas en la aplicación cliente en el servidor.

El subsistema Web brinda una interfaz de interacción con el resto de las aplicaciones del SCADA, de esta forma se logra una mayor integración y hace que esta nueva herramienta sea configurable para distintos tipos de procesos industriales.

10. Conclusiones

En capítulo se describieron los componentes desarrollado del lado del servidor, donde se selecciono a Apache como servidor Web por su estructura modular y la facilidad que brinda para añadirle nuevas funcionalidades. Se explicó el funcionamiento del Módulo de Transferencia de Daros (DTM) y la relación que tiene con el servidor HMI Web. Unas de las funcionalidades más importantes que tiene el servidor HMI Web es la interacción con los demás módulos del sistema SCADA y gestión de los elementos gráficos.

Conclusiones Generales

Con el desarrollo del módulo de visualización Web y sus componentes, se ha logrado obtener una herramienta capaz de reutilizar objetos visuales desarrollados en el SCADA, manteniendo una interfaz semejante a la aplicación de escritorio, anteriormente desarrollada. Ha permitido expandir las funcionalidades de los sistemas SCADA al servicio Web de Internet, utilizando al máximo las posibilidades que brinda este servicio. La aplicación ha permitido la supervisión de procesos industriales sobre navegadores Web estándar, lo cual posibilita su funcionamiento sobre una gran variedad de dispositivos con capacidad de navegación Web, como son dispositivos celulares, PDAs, computadoras personales sin requerimientos de software adicional, entre otros más.

El desarrollo de los distintos componentes del módulo de visualización, pueden ser reutilizados en otras aplicaciones de interacción con aplicaciones Web o alguna aplicación servidora.

Ha permitido a los usuarios tener una interacción directa con el sistema, a través los mandos de acción que permite la aplicación. Se ha creado una solución configurable con los restos de los módulos del SCADA, lo que permite que sea aplicable a una amplia gama de procesos industriales.

El módulo de visualización Web ha posibilitado una reducción de costos, puesto que se racionaliza el trabajo, se reduce el tiempo y el dinero dedicado al mantenimiento. Mayormente los equipos con baja capacidad de procesamiento son más baratos que las costosas salas de operaciones de un sistema SCADA.

Recomendaciones

El producto que se obtuvo como resultado de esta investigación está en desarrollo, utilizando diversos tipos de tecnologías, por su variedad de componentes. Las recomendaciones que a continuación se mencionan están separadas por los distintos componentes que se desarrollaron.

1. Recomendaciones en el cliente Web.

Integrar a la aplicación Web un framework de AJAX que permita usabilidad añadida a la aplicación siempre teniendo en cuenta la potabilidad.

Crear un mecanismo de para repintado en el cliente, para actualización de cada imagen por separado y no de forma conjunta.

2. Recomendaciones para el módulo de transferencia de datos (DTM).

Se recomienda crear una nueva funcionalidad que permita al módulo conocer la autenticación de cada pedido.

Crear un mecanismo de seguridad utilizando este módulo y el servidor de Seguridad del SCADA para la aplicación Web.

Se recomienda el uso de este módulo para la comunicación entre el servidor Web Apache y una aplicación en C++.

3. Recomendaciones para el servidor HMI Web.

Implementar las funcionalidades necesarias para comunicar el servidor HMI Web con el servidor de Seguridad del SCADA.

Desarrollar los sinópticos gráficos nuevos que se adicionen al SCADA, para tener una aplicación mas completa.

Implementar los sumarios de alarmas, tan importante utilidad en los sistemas SCADA.

Referencias bibliográficas

[1] **Acevedo Sánchez, Jose. 2006.** *Instrumentación y Control Básico de Procesos.*, Díaz de Santos, 2006.

[2] **Acevedo Sánchez, Jose. 2002.** *Control Avanzado de Procesos.*, Díaz de Santos, 2002.

[3] **Roca Cusido, Alfred.** *Control de Procesos*, 2000.

[4] **Rodríguez Penin, Aquilino. 2006.** *Sistemas SCADA*, 2006.

[5] **Apache HTTP Server Project**, <http://httpd.apache.org/>

[6] **Apache**, <http://httpd.apache.org/docs/2.0/programs/apxs.html>

Anexos

Glosario de Términos

Sinópticos: Que toma la forma de un objeto.

Despliegues: Exhibición, demostración de una actividad.

Cookies: Es un fragmento de información que se almacena en el disco duro del visitante de una página Web a través de su navegador, a petición del servidor de la página.

Metadato: Se refiere a un grupo de datos, llamado recurso, que describen otros datos.

API: Una interfaz de programación de aplicaciones. Con un conjunto de funciones o procedimientos que ofrece una biblioteca para ser utilizado por otro software.

Biblioteca: Es un conjunto de subprogramas para desarrollar software.

SVG: Es un lenguaje para describir gráficos vectoriales, tanto estáticos como animados.

PNG: Es un formato gráfico en un algoritmo de compresión sin pérdida para bitmaps no sujeto a patentes.

Hardware: Corresponde a todas las partes físicas y tangibles de una computadora

Software: Se refiere al equipamiento lógico o soporte lógico de un computador digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema.

Gráficos de tendencias: Es un patrón de comportamiento de los elementos de un entorno particular durante un periodo de tiempo.

Sensor: Es un dispositivo capaz de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas.

Objeto iframe: Es un objeto independiente incrustado en el cuerpo de una página Web.

Módem: Es un dispositivo que sirve para modular y desmodular (en amplitud, frecuencia, fase u otro sistema) una señal llamada portadora mediante otra señal de entrada llamada moduladora.