Universidad de las Ciencias Informáticas
Facultad 5

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Título: "XStormClient, visualizador web para el SCADA Guardián del Alba."

Autor: Yuri VictorMunayev Patiño

Tutor: Ing. Ernesto Leyva Barrero.
Co-Tutor: Ing. Adisleidys Mirabal Pérez.

Ciudad de la Habana, 2011

DECLARACIÓN DE AUTORÍA

Por este medio se declara que soy él únicos autores de este trabajo y se autoriza a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estime pertinente con este trabajo.
Para que así conste se firma la presente a los __ días del mes de _____ del 2011.

_____ _____
Firma del Autor Firma del Tutor
(Yuri VictorMunayev Patiño) (Ernesto Leyva Barrero)

_____
Firma delCo-Tutor
(Adisleidys Mirabal Pérez)

Agradecimientos

Resumen

En los últimos años la automatización de procesos industriales se ha convertido en un gran avance para el sector empresarial. Dentro de las aplicaciones informáticas que se utilizan con este fin, se encuentran lossistemas de supervisión, control y adquisición de datos, conocidos como SCADA. En la actualidad una gran parte de estos sistemas han enfocado su diseño hacia aplicaciones Web, pues esta permiten mayor extensibilidad, accesibilidad y estandarización.

Este trabajo surge a partir de la necesidad de los directivos, supervisores y personal autorizado de la empresa PDVSA, de visualizar la información del sistema SCADA Guardián del ALBA, en dispositivos portátiles y estaciones de trabajo que no tengan instalado dicho sistema. El objetivo de esta investigación es desarrollar una aplicación web capas de visualizar la información del sistema SCADA Guardián del ALBA. Para cumplir esta meta se describe una solución partiendo del diseño de software y se detalla la implementación del sistema.

Palabras Claves: Visualización, comunicación, web y SCADA.

Índice de Contenido

Introducción

Hace unos añosera casi imposible pensar que la humanidad alcanzaría el desarrollo tecnológico que posee hoy en día. En poco tiempolas Tecnologías de la Información y las Comunicaciones (TIC) han evolucionado considerablemente, hasta el punto de volverse casi imprescindible para algunos sectores de nuestra sociedad. La automatización de los procesos industrialesse ha convertido en un gran avance para el sector empresarial, con su uso, se han revolucionado las producciones a gran escala, reduciendo el peligro de catástrofes, además deobtener grandes ahorros en tiempo, dinero y recursos.

Nuestro país no ha quedado exento de este desarrollo y desde hace algunos años se trabaja en la automatización de los procesos de las industrias, con el objetivo de mejorar la productividad ycalidad de los productos. La UCI(Universidadde las Ciencias Informáticas),unade las pioneras en este sentido,tiene un peso importante dentro del sistema científico cubano, lo que impone a este centro una participación destacada en las investigaciones dirigidas a buscar soluciones a los problemas que se presentan en el proceso de automatización.En su Facultad 5 radica el CEDIN(Centro de Informática Industrial), donde se desarrollan varios proyectos que utilizan los Sistemas de Control Industrial como solución a estos problemas, entre ellos se encuentra SCADA Guardián del ALBA,que se ha desarrollado como una alternativa de soberanía tecnológica para la empresa venezolana PDVSA (Petróleos de Venezuela S.A).

El SCADA está compuesto por varios módulos, los más importantes son Visualización, Adquisición, Histórico y Seguridad, todos conectadosa través del módulo de Comunicación. El módulo de Visualización a su vez está formado por dossubmódulos: Reportes y HMI (Human Machine Interface). HMI es una interfaz deusuario, que permite la comunicación entre los usuarios y el sistema. Dentrode HMI se encuentran dos Aplicaciones principales: ambiente de configuración y ambiente de ejecución. El ambiente de configuración se utiliza para configurar todos los módulos del SCADA. El ambiente de ejecución es el encargado de visualizar, a través de la pantalla, los datos recolectados de los dispositivos de campo, para su supervisión y control.

En la actualidades muy engorroso para los usuarios utilizar el ambiente de ejecución o visualizador, pues este depende del sistema operativo Debian6.0,se necesita tener instalado la mayor parte del SCADA en los ordenadores, además para realizar modificaciones se debenactualizar todos los clientes por separado.

Provocando que los directivos, supervisores y personal autorizado de la empresa PDVSA, no puedan visualizar la información del sistema SCADA Guardián del ALBA,en dispositivos portátiles y estaciones de trabajo que no tengan instalado dicho sistema. Estas condiciones, traen como consecuencia que para desarrollar la supervisión y el análisis de la producción y equipamiento, sea necesario la compleja instalación del sistema SCADA, así como el traslado hasta las distintas estaciones de control, muchas de las cuales se encuentran apartadas de las entidades administrativas, provocando gastos innecesarios y molestias al personal.

Teniendo en cuenta lo expuesto anteriormente se define como **problema científico** la siguiente interrogante:

¿Cómo visualizar la información del sistema SCADA Guardián del ALBA, en dispositivos portátiles y estaciones de trabajo, que no tengan instalado dicho sistema?

Según el problema científico se plantea como**objeto de estudio**: Visualización de la información de sistemasSCADAs.

Definiendo como **objetivo general** de la investigación: Desarrollar una aplicación web para visualizar la informacióndel sistema SCADA Guardián del ALBA.

Teniendo en cuenta la relación entre el problema, el objeto de estudio y el objetivo general de lainvestigación se define como**campo de acción**: Visualización de la información del sistema SCADA Guardián del ALBA en la web.

Como **idea a defender**se define: La creación del visualizador XStormClient, permitirá a directivos, supervisores y personal autorizado de la empresa PDVSA, visualizar la información del sistema SCADA Guardián del ALBA, desde dispositivos portátiles y estaciones de trabajo,que no tengan instalado dichosistema, utilizando un navegador web.

Para guiar el desarrollo de esta investigación, se proponen las siguientes tareas:

- Caracterización del sistema SCADA Guardián del ALBA a partir del estudio de la bibliografía referente a su funcionamiento.

- Estudio del estado del arte, de aplicaciones web utilizadas en la visualización de sistemas SCADA, para realizar una comparación acerca de la tecnologías usadas.
- Estudio de protocolos de comunicación con tecnologías Web para optar por el más conveniente de acuerdo con las necesidades de la investigación.
- Estudio de diferentes bibliotecas y framework de desarrollo de aplicaciones Web en el lado cliente (client-side), para seleccionar la que se ajuste al objetivo del trabajo.
- Diseño de la interfaz de la aplicación para la visualización del sistema SCADA Guardián del ALBAen la web.
- Generación de artefactos propuestos por la metodología de investigación.
- Implementación de una aplicación para la visualización del sistema SCADA Guardián del ALBAen la web.
- Realización de pruebas para lavalidación del software.

Durante el desarrollo de esta investigación, se hace uso de varios **Métodos Teóricos** y **Empíricos** para profundizar en algunos temas y dar cumplimiento a las tareas antes mencionadas:

Se utilizan **Métodos Teóricos**, para obtenerconocimientos sobre el estado del arte de los procesos que se quieren automatizar y su relación con otros procesos.

Método histórico-lógico: Para la comprensión de los antecedentes y las tendencias actuales de los sistemas SCADAweb.

Método analítico-sintético: Para la definición de las funcionalidades necesarias del visualizador web del SCADAGuardián del Alba.

Los **Métodos Empíricos** por otra parte, permiten extraer de los procesos analizados las informaciones que se necesitan sobre ellos, a través de observaciones y del uso de otras técnicas de recopilación de la información.

Observación: Permitirá observar cómo se comporta la visualización y actualización del visualizador web del SCADA Guardián del Alba.

Experimento: Para la elaboración del visualizador de despliegues en la web del SCADA Guardián del Alba. Además para realizar una comparación entre diferentes técnicas y tecnologías de comunicación de aplicaciones web.

Criterio de expertos: Para adquirir conocimientos referentes al tema y continuar con la investigación, mediante encuestas o entrevistas, cuya experiencia y opiniones pueden ser de una valiosa contribución para el desarrollo del trabajo.

Con la culminación de esta investigación, se espera obtener una aplicación capaz de visualizar la información del sistema SCADA Guardián del ALBA en un entorno web, que pueda ser utilizada en las instalaciones de PDVSA, con diferentes objetivos, como supervisar la producción, el estado del equipamiento por parte del personal de mantenimiento y el análisis y supervisión desde las entidades administrativas.

En este trabajo de diploma se propone estructurar los capítulos de la siguiente forma:

Capítulo 1-Fundamentación teórica
Se desarrolla la fundamentación teórica del presente trabajo.

Capítulo 2-Construcción de la solución
Se detallarán los artefactos de la metodología que se propone en el capítulo anterior, quedarán definidaslas tareas generadas a partir del desarrollo de las funcionalidades y los diseños de casos de pruebas para validar la solución. Además se diseña un sistema de clases, en correspondencia con las técnicas de la programación, que luego son implementadas teniendo como resultado final un prototipo funcional del sistema propuesto.

Capítulo 3-Pruebas
Se realizan pruebas al producto para comprobar el buen funcionamiento del mismo.

Capítulo 1: Fundamentación Teórica

Introducción

En este capítulo se explican los módulos y funcionamiento de un SCADA para la gestión y control de datos, enfatizando en el módulo HMI. Además se describen las principales tecnologías que se utilizaron para construir el software, incluyendo lenguaje de programación, framework, y las herramientas de desarrollo empleadas.

1.1 Generalidades de un sistema SCADA

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 un hardware de señal de entrada y salida, sensores y actuadores, controladores, HMI (Interfaz Hombre Máquina), redes, comunicaciones, bases de datos, entre otros. Los sistemas SCADA mejoran la eficacia del proceso de monitoreo y control proporcionando la información oportuna para poder tomar decisiones operacionales apropiadas. De igual forma, al contar con información (alarmas, históricos, paradas, etc.) de primera mano de lo que ocurre u ocurrió en el proceso, permite la integración con otras herramientas del negocio como lo son intranets, ERP, entre otras(1).

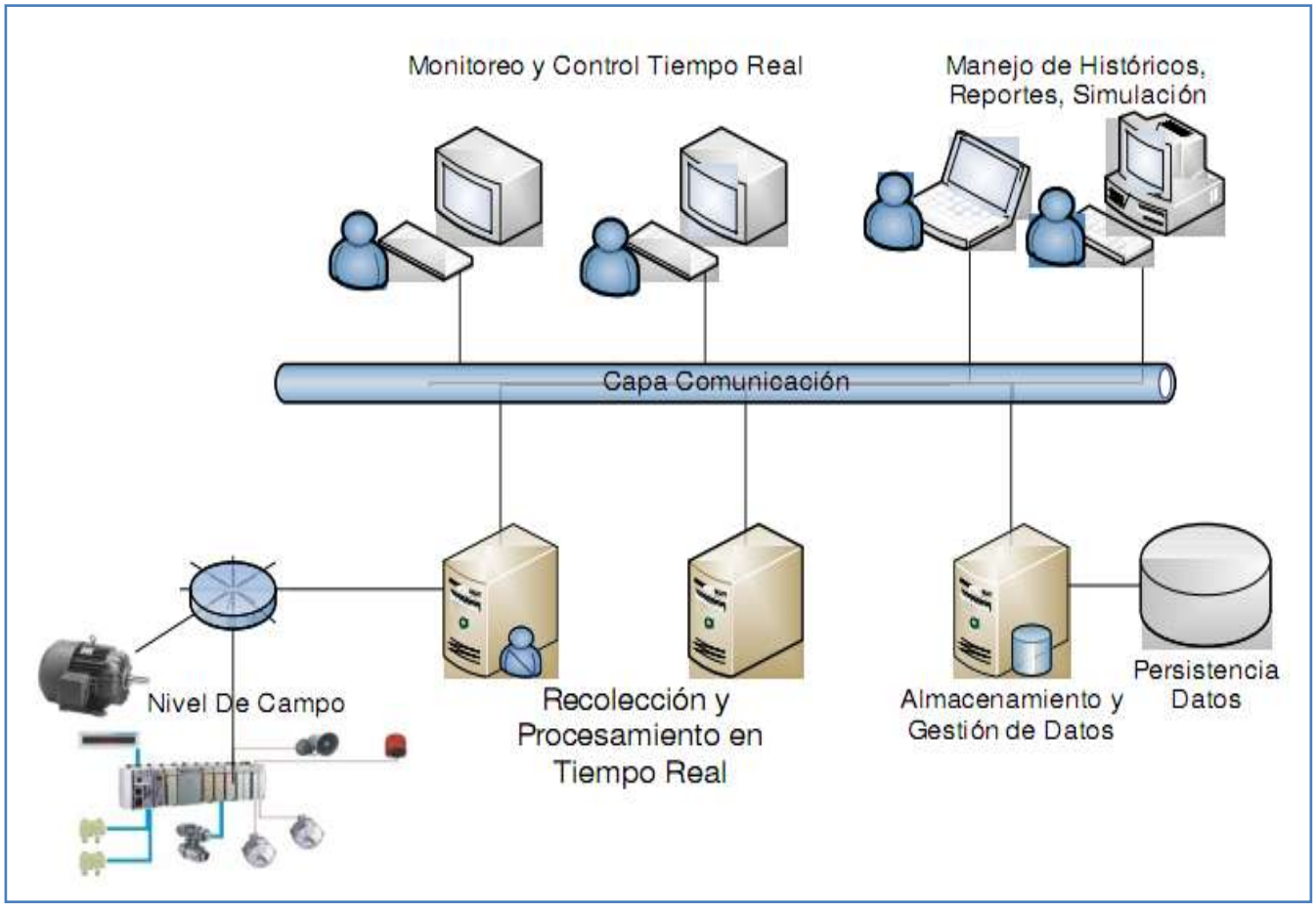


Figura. 1Ejemplo del esquema de un SCADA.

Un SCADA debe cumplir tres **funciones principales:**

Adquisición de datos: significa que el sistema tiene la capacidad de obtener información desde el sistema de control, por ejemplo, sobre valores de temperatura o presión y a diferencia de la supervisión los datos son registrados o almacenados para su posterior explotación.

Supervisión: significa poder observar o monitorear aquello que sucede en el proceso industrial, el equipo o la maquinaria, por ejemplo, conocer (generalmente de una forma gráfica) si un motor se encuentra encendido o apagado.

Control: significa tener la posibilidad de ejecutar comandos; en otras palabras poder enviar instrucciones hacia el sistema.

El usuario, mediante herramientas de visualización y control, tiene acceso al procesador digital, generalmente un ordenador donde reside la aplicación de supervisión y control. La comunicación entre estos dos sistemas se suele realizar a través de redes de comunicación corporativas. El procesador digital capta el estado del sistema a través de los elementos sensores e informa al usuario a través de la herramienta HMI. Basándose en los comandos ejecutados por el usuario, el procesador digital inicia las acciones pertinentes para mantener el control del sistema a través de los elementos actuadores.

Funciones más específica de un SCADA:

- Transmisión de información con dispositivos de campo.
- Gestión de datos con bajos tiempos de acceso.
- Representación gráfica de los datos. Interfaz del Operador o HMI.
- Explotación de los datos adquiridos para gestión de la calidad, control estadístico, gestión de la producción y gestión administrativa y financiera.

Prestaciones de un SCADA

El paquete SCADA, comprende toda una serie de funciones y utilidades encaminadas a establecer una comunicación lo más clara posible entre los procesos y el operador, entre las prestaciones de una herramienta de este tipo destacan:

- **Monitorización:** Representación de datos en tiempo real a los operadores de la planta. Se leen los datos de los autómatas (temperatura, velocidad, detectores). Una máquina simple, una hidroeléctrica, un parque eólico, pueden ser vigilados desde muchos kilómetros de distancia.
- **Visualización de los estados de las señales del sistema (alarmas y eventos):** Reconocimiento de eventos excepcionales acaecidos en la planta y su inmediata puesta en conocimiento a los operarios, para efectuar las operaciones correctas pertinentes. Además los paneles de alarmas pueden exigir alguna acción de reconocimiento por parte de los operarios, de forma que quede registrada la incidencia.

- **Mando:** Posibilidad de que los operadores puedan cambiar consignas u otros datos clave del proceso, directamente desde el ordenador (marcha, paro, modificación de parámetros). Se escriben datos sobre elementos de control.
- **Seguridad de los datos:** Restringiendo zonas de programa comprometidas a usuarios no autorizados, registrando todos los accesos y acciones llevadas a cabo por cualquier operador.
- **Programación numérica:** Permite realizar cálculos aritméticos de elevada resolución sobre la CPU de ordenador.

1.2 Módulos de un SCADA

- **Bases de Datos Históricos:** Es el encargado de almacenar la información recibida desde el campo, así como la sucesión de alarmas y eventos**.** Esta información esde vital importancia para realizar cualquier tipo de análisis posteriores comodiagnósticos o reportes.
- **Middleware:**Este módulo representa la capa de software encargada de la comunicación entre los diferentes procesos distribuidos, de mediano y alto nivel.
- **Seguridad:** Permite a los usuarios autenticarse en el sistema, y de esta forma poder acceder sólo a los recursos que tiene asignado su rol. Posee herramientas para la protección ante ataques piratas, fallos eléctricos, problemas de red, entre otros.
- **Procesamiento de Datos:** Este es el "centro neurológico" del sistema, el cual supervisa y recoge la información delresto de las subestaciones, bien sean otros ordenadores conectados (ensistemas complejos) a los instrumentos de campo o directamente sobre dichosinstrumentos, algunas de las funciones que cumple son:
  - Interrogar de forma periódica a las RTU, y transmitirle consignas; siguiendo usualmente un esquema maestro-esclavo.
  - Actuar como interfaz al operador, incluyendo la presentación de información de variables en tiempo real, la administración de alarmas, y la recolección y presentación de información historiada.
  - Puede ejecutar software especializado que cumple funciones específicas asociadas al proceso supervisado por el SCADA. Por ejemplo, software para detección de pérdidas en un oleoducto.
- **HMI:**Es el aparato que presenta los datos a un operador(2).

1.3 El módulo Interfaz Hombre-Máquina

La Interfaz Hombre-Máquina o HMI ("Human Machine Interface") es el aparato que presenta los datos a un operador (humano) y a través de la cual éste controla el proceso. La industria de HMI nació esencialmente de la necesidad de estandarizar la manera de monitorear y de controlar múltiples sistemas remotos, PLC y otros mecanismos de control. Aunque un PLC realiza automáticamente un control pre-programado sobre un proceso, normalmente se distribuyen a lo largo de toda la planta, haciendo difícil recoger los datos de manera manual, los sistemas SCADA lo hacen de manera automática.
Históricamente los PLC no tienen una manera estándar de presentar la información al operador. La obtención de los datos por el sistema SCADA parte desde el PLC o desde otros controladores y se realiza por medio de una red, posteriormente esta información es combinada y formateada. Un HMI puede tener también vínculos con una base de datos para proporcionar las tendencias, los 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. Desde cerca de 1998, virtualmente todos los productores principales de PLC ofrecen integración con sistemas HMI/SCADA, muchos de ellos usan protocolos de comunicaciones abiertos y no propietarios. Numerosos paquetes de HMI/SCADA de terceros ofrecen compatibilidad incorporada con la mayoría de PLC, incluyendo la entrada al mercado de ingenieros mecánicos, eléctricos y técnicos para configurar estas interfaces por sí mismos, sin la necesidad de un programa hecho a medida escrito por un desarrollador de software(3).

Aplicaciones que componen un HMI:

- **Ambiente de Configuración:** Esta aplicación permite configurar varios procesos o partes de ellos, posibilitando 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, se seleccionan los drivers de comunicación que permitirán el enlace con los elementos de campo y la conexión o no en red de estos últimos. Estas configuraciones son las que más tarde utilizará el visualizador para poner en ejecución la representación gráfica de la planta, fábrica o lugar donde ha sido instalado.

- **Ambiente de Ejecución:** Proporciona al operador las funciones de control y supervisión de la planta. El proceso a supervisar se representa mediante sinópticos gráficos que cambian dinámicamente a diferentes formas y colores, según los valores leídos en la planta o en respuesta a las acciones del operador. Las funcionalidades principales con la que debe contar toda aplicación de este tipo son:

  - **Visualización de despliegues:** Es el encargado de mostrar de forma gráfica los procesos que se siguen en una planta. Esto se realiza a través de objetos gráficos que suelen dividirse en dos grupos, simples y complejos, los primeros se caracterizan por ser objetos estáticos, como circunferencias, líneas, imágenes, y los complejos se caracterizan por variar su estado en el transcurso del tiempo según los datos leídos en la planta, entre los principales se encuentran las gráficas de tendencia, los relojes y las reglas.
  - **Visualización de alarmas:** Es una interfaz gráfica que concentra las condiciones de procesos en críticas, medias y bajas presentes en el sistema y cuyo objetivo primordial es guiar al operador a la detección del origen de la falla y supervisar la ejecución de las medidas de corrección automatizada o manual.
  - **Visualización de puntos:** Es la interfaz gráfica encargada de visualizar los detalles de un punto, tales como: nombre, calidad, valor, fecha, entre otros valores.
  - **Visualización de eventos:** Muestra las acciones que han sido inducidas en el sistema.

Figura. 2Ejemplo del Zenon HMI SCADA.

Como se observa, los visualizadores de hoy en día ofrecen a los operadores las más sofisticadas técnicas de supervisión y control, brindado la posibilidad de simular partes de la planta con los gráficos sinópticos, haciendo sumamente intuitivo la operación de una planta específica.

1.4 Interfaz Hombre Máquina en la Web

En la actualidad, el acelerado crecimiento de los sistemas de comunicación ha hecho de la Internet una tecnología portadora de gran variedad de servicios. Ha permitido que el acceso a estos servicios pueda realizarse desde gran variedad de dispositivos: teléfonos móviles, 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.

Estudio del estado del arte sobre sistemas SCADA con HMI en la Web.

A continuación se relacionan los principales sistemas SCADA que cuentan con un HMI en la Web (Integraxor, CSWorks, NubitekFalcon).

Tabla. 1 Resumen características de sistemas SCADA con HMI Web.

Software	Integraxor	CSWorks	NubitekFalcon
Fabricante	EcavaSdnBhd's	CSWorks Inc.	Nubitek
Marco de desarrollo	Javascript	Microsoft Silverlight	Prototype
Plataforma de desarrollo del lado del servidor	Microsoft .NET	Microsoft .NET	Microsoft .NET
Control de Usuarios	Sí	Sí	Sí
Gestión de Alarmas	Sí	Sí	Sí
Representación de Gráficos	SVG	SVG	GIF
Método de comunicación	AJAX	AJAX	AJAX

En la tabla se pueden observar las similitudes y diferencias que poseen. Es importante recalcar el uso del estándar SVG, para la representación de los sinópticos gráficos por parte de Integraxor y CSWorks, también el uso de AJAX para comunicarse con el servidor. Todos son sistemas propietarios basados en tecnologías de Microsoft.

1.5 Tendencias y Tecnologías

En este epígrafe se realiza un análisis de las tendencias y tecnologías actuales en el campo del desarrollo de aplicaciones web. La solución se divide en dos partes fundamentales, el apartado de comunicación y el apartado de visualización.A continuación se explica el estudio realizado a dichos apartados.

1.5.1 Comunicación

Debido a que las comunicaciones es una parte importante en el desarrollo de la solución. En este apartado se hace un análisis de las principales técnicas y tecnologías de comunicación para aplicaciones web. Entre las que se encuentran:

AJAX

El artículo"Ajax: A New ApproachtoWebApplications" publicado por Jesse James Garrett el 18 de Febrero de 2005 define AJAX de la siguiente forma: "Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes".

En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puedetraducir como "JavaScript asíncrono + XML".

Las tecnologías que forman AJAX son:

- XHTML, HTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías(4).

XMPP

Protocolo extensible de mensajería y comunicación de presencia (XMPP), es una tecnología libre para la comunicación en tiempo real, que usa XML 5 como el formato base para el intercambio de información. En esencia, XMPP provee la forma de enviar pequeñas porciones de XML de una entidad a otra en el menor tiempo posible.

Aun cuando XMPP fue desarrollado originalmente en la comunidad de código abierto Jabber, el protocolo en sí no es un proyecto de código abierto como Apache, sino más bien un estándar abierto como HTTP. Como resultado, XMPP es una tecnología abierta que no está ligado a ningún proyecto de software o empresa. Las especificaciones XMPP definen protocolos abiertos que se usan para la comunicación entre

entidades de la red. XMPP define los protocolos y formatos de datos que alimentan en tiempo real las interacciones a través de Internet(5).

Server-SentEvent (SSE)

El protocolo HTTP está considerado como un modelo de petición-respuesta, lo que significa que el cliente envía una solicitud HTTP y espera hasta que la respuesta HTTP se reciba. Por lo tanto, normalmente el servidor no puede comunicarse con el cliente a menos que sea solicitado. HTML5 proporciona una forma nativa para controlar los eventos enviados del servidor, define la API Server-SentEvent (SSE) para la apertura de una conexión HTTP, para recibir notificaciones enviadas desde un servidor.

Su funcionamiento comienza cuando el cliente abre una secuencia de eventos, mediante la creación de un EventSource, que tiene una dirección URL de origen de evento como parámetro. El controlador de eventos onMessage se llamará cada vez que el origen genere unnuevo dato. Como AJAX, que se pueden comunicar de forma asíncrona desde el cliente al servidor y, ahora con SSE se puede hacer en la dirección opuesta, desde el servidor al cliente, una vez más de forma asincrónica.

Resumiendo SSE es una especificación del estándar HTML5 orientada a crear un protocolo de comunicación unilateral, mediante el cual se envían datos desde el servidor al cliente(6).

WebSocke

Especificación del estándar HTML5, orientada a crear un protocolo de comunicación bilateral, mediante el cual se envían datos desde el servidor al cliente Web y viceversa.

La especificación de HTML5 WebSockets, define la API de sockets que permite a las páginas web usar el protocolo Secure Socket web, para comunicación bidireccionalcon un host remoto. Se presenta la interfaz WebSocket y define un canal de comunicación bidireccional, que opera a través de una sola toma en la Web. HTML5 Web Sockets pone a disposición del programador una conexión de socket, a través de Internet, con una sobrecarga mínima. Proporcionando una enorme reducción del tráfico innecesario en la red y disminuyendo la latencia. Estas soluciones se utilizan a menudo para empujar datos en tiempo real a los clientes, o incluso simular una conexión bidireccional, para mantener dos conexiones HTTP.

Para conectarse desde cliente web a un servidor utilizando HTML5 Web Sockets, se crea una instancia nueva WebSocket y se le pasa la dirección URL del servidor. La especificación define un ws:// y un wss ://, esquemas para indicarlas conexiones WebSocket y las conexiones WebSocket seguras, respectivamente. Una conexión WebSocket se establece mediante la actualización del protocolo HTTP, al protocolo Secure Socket Web, durante la conexión inicial entre el cliente y el servidor, sobre la misma conexión subyacente TCP/IP(7).

Estudio

Para la selección de una de las diferentes técnicas y protocolos de comunicación estudiados, se realizó un estudio comparativo, teniendo en cuenta las características que eran críticas para el desarrollo de la aplicación como: tiempo real, seguridad, sentido de la comunicación y concurrencia. A continuación se exponen los resultados obtenidos.

Cabe destacar que la característica más crítica es el tiempo real, o sea, el tiempo en que podemos enviar distintas secuencias de datos para el refrescamiento de las variables y alarmas. Existen dos tipos de implementaciones fundamentales de tiempo real. Están por una parte los duros o de hardware, que son muy eficientes pero costosos y complejos de implementar, y los suaves o de software que deben cumplir el requisito de ser deterministas, o sea, que el sistema tome un tiempo previamente determinado para realizar un conjunto de operaciones. Para esto se recurre a la experimentación, uno de los métodos empíricos más usados. El experimento consiste en el envío de 150punto, cada X segundos, para percibir a partir de cuál segundo comienza a presentarse latencia.

Tabla. 2Ejemplo de experimento realizado a la técnica AJAX.

Protocolo	Cantidad de Puntos	Tiempo en segundos	Presencia de latencia
AJAX	150	6	NO
AJAX	150	3	SI
AJAX	150	5	NO
AJAX	150	4	SI
AJAX	150	4.5	NO

En la tabla se observa que el tiempo de respuesta sin latencia para la cantidad de puntos deseada,es a partir de 4segundos en adelante.

Con los protocolos WebSocket, SSE y XMMP se procede de la misma manera, arrojando los siguientes datos.

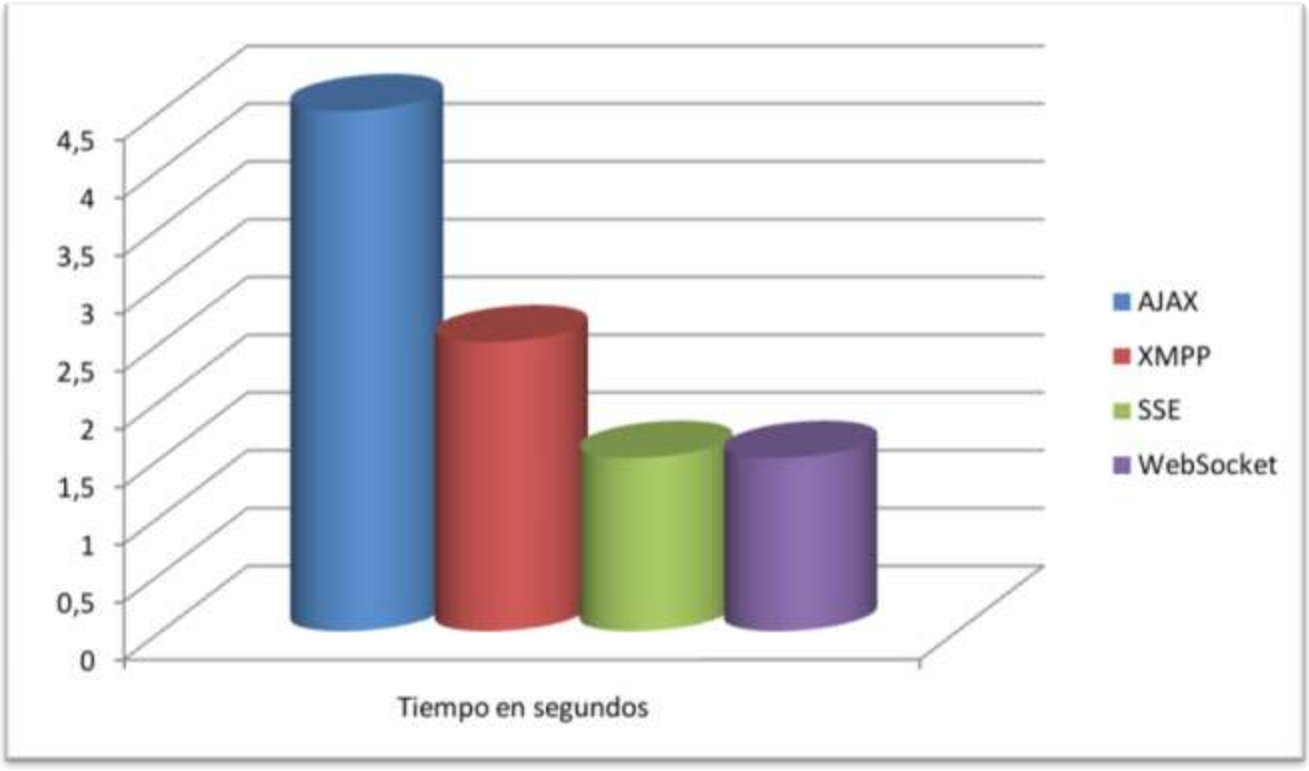


Figura. 3 Comparaciones de capacidad de enviar los datos requeridos en el menor tiempo.

En cuanto al envío de datos, los protocolos que más se destacan son SSE y WebSocket, puesel tiempo de respuesta sin latencia para la cantidad de puntos deseada, se encuentra en el intervalo de 1,5 a 2 segundos, en cuanto a XMPP, acotar que su capacidad de enviar datos en el menor tiempo posible disminuye considerablemente, cuando la cantidad de datos aumenta.

Atendiendo a las características de cada uno de los protocolos estudiados es importante destacar que en cuanto a la seguridad, el protocolo más robusto es XMPP, ya que el mismo posee sistemas de seguridad como Simple Authentication and Security Layer (SASL) y TransportLayer Security (TLS) (3). En cuanto al sentido de las comunicaciones se queda por debajo SSE, dado que solo permite conexiones en unsolo

sentido, del servidor al cliente. Todas estas técnicas y protocolos permiten gran concurrencia.

Por lo antes expuesto, se puedellegar a la conclusión de que el protocolo WebSocket es el que más se ajusta al desarrollo de la solución, porque soporta el envío de grandes cantidades de datos en tiempo realcon grandes concurrencias, es bidireccional, se puedes hace peticiones cross-domain y posee una seguridad aceptable.

1.5.2 Visualización

En este apartado se realiza un estudio sobre las principales tecnologías estándares para la visualización web. Dentro de ellas se encuentran HTML, CCS y SVG.

HTML

El propio W3C[1] define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos, a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica(8).

HTML5 es la actualización de HTML, el lenguaje en el que es creada la web. HTML5 también es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de JavaScript.

La versión anterior y más usada de HTML, HTML4, carece de características necesarias para la creación de aplicaciones modernas basadas en un navegador. El uso fuerte de JavaScript ha ayudado a su mejora, gracias a frameworks como jQuery, jQuery UI, ExtJS, YUI 3, entre otros. Flash en especial ha sido usado en reemplazo de HTML para desarrollar aplicaciones web que superaran las habilidades de un navegador: Audio, video, webcams, micrófonos, datos binarios, animaciones vectoriales, componentes de interfaz complejos, entre muchas otras cosas. Ahora HTML5 es capaz de hacer esto sin necesidad de plugins y con una gran compatibilidad entre navegadores(9).

Para esta solución se utilizará HTML para crear los elementos de la interfaz visual, permitiendo de esta

[1] W3C consorcio internacional que produce recomendaciones para la World Wide Web.

manera que pueda ser visualizado en todo tipo de navegadores web.

CSS

CSS es un lenguaje de hojas de estilos, creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar los contenidos y la definición de su aspecto presenta numerosas ventajas, pues obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, entre otros(10).

CSS será utilizado para moldear los elementos de las interfaz, haciéndolos fáciles de visualizar por el usuario.

Scalable Vector Graphic (SVG)

Es un estándar de la W3C que define una gramática XML para describir gráficos de dos dimensiones. Cualquier programa, tales como un navegador que reconozca XML pueden mostrar la imagen usando la información proporcionada en el formato SVG. La parte del término “gráficos escalable” hace hincapié en que las imágenes SVG se pueden ampliar fácilmente, a diferencia de las imágenes especificadas en gráficos de mapa de bits. Así, el formato SVG permite la visualización de una imagen sobre una pantalla de ordenador de cualquier tamaño y resolución, ya sea una pequeña pantalla LCD en un teléfono celular o una gran pantalla CRT en una estación de trabajo. Además de la facilidad de la reducción del tamaño y la ampliación, SVG permite que los textos dentro de imágenes para ser reconocidas como tales, por lo que el texto puede ser localizado por un motor de búsqueda y fácilmente traducible a otros idiomas. Otra ventaja potencial sobre los formatos de imagen estándar de la web (GIF y JPEG) es su tamaño, en comparación con una imagen de mapa de bits, una imagen SVG puede ser mucho menor lo que

disminuye el tiempo de carga. Los gráficos SVG pueden ser interactivos y dinámicos, con animaciones que pueden ser definidas y activas de modo declarativo o por medio de scripts. Además de poder usar un script adicional con acceso al DOM [2]de SVG para obtener aplicaciones más complejas(11).

Se decidió utilizar SVG para la visualización de componentes gráficos más complejos, específicamente para los componentes gráficos que representan los diferentes dispositivos del Sistema SCADA Guardián del ALBA.

1.5.3 Lenguaje de Programación JavaScript

Se utilizó Javascript para programar tanto el apartado de comunicación, como el apartado de visualización. Javascript es un lenguaje de programación interpretado, se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Comenzó como una manera de manipular elementos de las páginas web (como imágenes y campos de formulario), pero ha crecido enormemente. Además de las secuencias de código del lado cliente, en estos días se puede usar JavaScript para programa en una creciente variedad de plataformas. Puede escribir código del lado servidor (usando. NET o Node.js), aplicaciones de escritorio (que funcionan en todos los sistemas operativos) y las extensiones de aplicación (por ejemplo, para Firefox o Photoshop), aplicaciones móviles, y los scripts de línea de comandos.

JavaScript es un lenguaje inusual. No tiene clases, y las funciones son objetos de primera clase que se utilizan para muchas tareas. En un principio el lenguaje fue considerado deficiente por muchos desarrolladores, pero en años más recientes, esto ha cambiado. Interesantemente, lenguajes como Java y PHP comenzaron a añadir características como los ámbitos (closures) y las funciones anónimas, que los desarrolladores de JavaScript han estado disfrutando y dando por sentado durante un rato(12).

1.5.4 Framework YUI 3.+

Para aumentar las potencialidades del lenguaje Javascript así como para mejorar el proceso de desarrollo se decidió utilizar YUI 3.+ como framework. YUI 3 es framework de código abierto de JavaScript y CSS para crear aplicaciones web enriquecidas einteractivas. Construido por los ingenieros de frontend de

[2]Modelo de Objetos de Documento

Yahoo, se ofrece bajo la licencia BSD [3]y está disponible en GitHub[4]para la bifurcación y la contribución.Dentro de sus principales características se encuentra las siguientes:

- Larapidez debido a que posee un núcleo ligero y su arquitectura modular, lo hace escalable, rápido y robusto.
- Tiene una API muy completa, intuitiva y bien documentada. Que puede ser utilizada para desarrollar tareas básicas como una simple manipulación del DOM y otras más complicadas como el desarrollo de aplicaciones web complejas.
- Es fácil de mantener en navegadores web, dispositivos móviles y servidores.
- Consta de una comunidad próspera, una infraestructura de atención a la arquitectura y un completo conjunto de herramientas, que ayudan a crear código como un profesional.
- Utiliza conceptos novedosos para este tipo de tecnologías como: la carga dinámica de ficheros, lo que facilita y agiliza el proceso de carga de las aplicaciones, la implementación de varios e importantes patrones de diseño, que permiten construir aplicaciones más robustas y con mayor rapidez y extensibilidad, además de brindar compatibilidad entre diferentes navegadores (cross-browser) lo que permite que las aplicaciones funcionen de igual manera en los distintos navegadores(13).

1.5.5 IDE de desarrollo

Todo el proceso de implementación se desarrolló sobre Aptana Studio 3,un IDE profesional de código abierto, que seutiliza para desarrollar y probar aplicaciones web en un único entorno.Su última actualización posee soporte para las especificaciones de las últimas tecnologíaspara esta plataforma, como HTML5, CSS3, JavaScript, Ruby, Rails, PHP y Python.

Además de incluir información sobre el nivel de apoyo para cada elemento, en los navegadores web más importantes.

1.6 Metodología de Desarrollo

[3]Berkeley Distribución de Software
[4]Repositorio de código de Internet

Las metodologíastradicionaleshan estado presentes durante mucho tiempo. No se han distinguido precisamente por ser muy exitosas, aún menos por su popularidad. La crítica más frecuente a estas metodologías es que son burocráticas. Hay tanto que hacer para seguir la metodología, que el ritmo entero del desarrollo se retarda.

Como una reacción a estas metodologías, un nuevo grupo ha surgido en los últimos años. Durante algún tiempo se conocían como ligeras, pero el término aceptado ahora es metodologías ágiles. Para mucha gente, el encanto de estas, es su reacción a la burocracia de las metodologías monumentales. Estos nuevos métodos buscan un justo medio entre ningún proceso y demasiado proceso, proporcionando simplemente suficiente proceso para que el esfuerzo valga la pena.

El resultado de todo esto es que los métodos ágiles cambian significativamente algunos de los énfasis de los métodos ingenieriles. La diferencia inmediata es que son menos orientados al documento, exigiendo una cantidad más pequeña de documentación para una tarea dada. En muchas maneras son más bien orientados al código: siguiendo un camino que dice que la parte importante de la documentación es el código fuente(14).

Dentro de estas metodologías una de las más usadas es XP (Programación Extrema), centrada en satisfacer al cliente y potenciar al máximo el trabajo en equipo. Se deriva en seis fases que se mencionan a continuación:

I. Exploración.
II. Planificación de la Entrega (Release).
III. Iteraciones.
IV. Producción.
V. Mantenimiento.
VI. Muerte del Proyecto.

Presenta los siguientes roles:

Administrador (programador): Su misión es administrador el ambiente de programación. Realiza las actividades de Configurar el Ambiente de Programación.

Programador: Su misión es ser responsable de implementar el código para dar soporte a los requisitos del cliente. Realiza las actividades de Definir el Estimado de la Tarea, Estimado del Requisito del Cliente, entre otros, generando los artefactos de XP construcción, XP pruebas unitarias.

Probador: Su misión es ayudar al cliente a definir y escribir pruebas de aceptación para los requisitos. Realiza las actividades de Correr las Pruebas del cliente, Automatizar las Pruebas del cliente y Configurar el Ambiente del Probador.

Cliente: Su misión es ser responsable de definir cuál es el producto correcto a construir, determinar las características del mismo y asegurarse de que el producto realmente satisface sus requerimientos. Realiza las actividades de Ajustar las iteraciones de las actividades, Definir la Iteración, Definir la Visión del documento, Escribir los requisitos, entre otros, generando los artefactos de Requisitos, Prueba del Cliente, Plan de Iteraciones, entre otros.

Encargado de Seguimiento: Su misión es medir y comunicar el progreso del proyecto. Realiza las actividades de Seguimiento del Progreso de la Iteración y Seguimiento del Progreso de Entrega del producto.

Entrenador (coaching): Su misión es ayudar a mantener la disciplina y aprendizaje del equipo. Realiza las actividades de Explicar Proceso, Mejorar las habilidades del Equipo, Resolver Conflictos, entre otros.

Esta metodología define un procesoiterativo e incremental, utilizar la programación en pareja, fomenta la reutilización de código y genera muy pocos artefactos, lo que acelera el proceso de terminación del software. Para proyectos de pequeña envergadura estas características resultan ventajas importantes(15). Por todo lo anterior se decidió escoger esta metodología para el desarrollo de este software.

1.7 Visual Paradigm.

Visual Paradigm para UML (UnifiedModelingLanguage, Lenguaje Unificado de Modelado), es una herramienta CASE (ComputerAided Software Engineering, Ingeniería de Software Asistida por Computadora)profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Los sistemas de modelado UML ayudan a una más rápida construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los

tipos de diagramas de clases, ingeniería inversa, generar código a partir de diagramas, generación de objetos a partir de bases de datos, generación de bases de datos a partir de diagramas de entidad relación y generar documentación, posee licencia gratuita y comercial Permite interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse). Se integra con Visio. Posee una plataforma, llamada SDE, capaz de integrarse con Eclipse, NetBeans, Oracle JDeveloper, JBuilder, IntelliJ IDEA, WebLogicWorkshop, Microsoft Visual Studio, entre otras.

Consideraciones parciales

En este capítulo queda demostrado que se debe desarrollar una aplicación web, capaz de visualizar los datos del sistema SCADA Guardián de ALBA, pues con su uso se permitirá a los directivos, supervisores y personal autorizado de la empresa PDVSA, tener acceso a la información del sistema, en dispositivos portátiles fuera de la planta y/o estaciones de trabajo, que no tengan instalado dicho sistema. Se utilizarácomo tecnologías web para la visualización HTML5, CCS3, SVG y Javascript,ypara la comunicaciónWebsocket, todas agrupadas en el potenteframework de desarrollo YUI 3.x. Comoherramienta de modeladoVisual Paradigm, porque se caracteriza por ser intuitiva para usuarios que no poseen la costumbre de trabajar con UML. Todo este procesoserá guiado por la metodología de desarrollo XP.

Capítulo 2: Construcción de la solución

Introducción

El objetivo del siguiente capítulo es describir el proceso de construcción de la solución, siguiendo la metodología de desarrollo XP. Se describen brevemente las principales funciones y el flujo actual de los procesos involucrados en el problema en cuestión, haciéndose un análisis de cómo se visualizan los datos en el sistema SCADA Guardián del ALBA.

Se muestran las historias de usuarios que fueron escritas por el cliente y se definen dos iteraciones para el desarrollo, elaborándose el Plan de Entrega para cada versión. Se definen las tareas generadas a partir del desarrollo de las historias de usuarios, durante las dos iteraciones planificadas.

2.1. Descripción de las acciones vinculadas al campo de acción

Actualmente los directivos, supervisores y personal autorizado de la empresa PDVSA, no pueden visualizar la información del sistema, en dispositivos portátiles yestaciones de trabajo que no tengan instalado un sistema SCADA. Estas condiciones, traen como consecuencia que para desarrollar la supervisión y el análisis de la producción y equipamiento, sea necesario la compleja instalación del sistema SCADA, así como el traslado hasta las distintas estaciones de control, muchas de las cuales se encuentran apartadas de las entidades administrativas, provocando gastos innecesarios y molestias al personal. Teniendo en cuenta que las tecnologías web brindan gran accesibilidad y extensibilidad, se puede desarrollar una aplicación capaz de visualizar la información del sistema SCADA Guardián del ALBA desde casi cualquier parte y en una amplia gama de dispositivos y sistemas operativos.

2.1.1. Propuesta del sistema

La realización de este trabajo está orientada al desarrollo de unaaplicación web, capaz de conectarse con la capa de comunicación del SCADAGuardián del ALBA, a través de XStormServer, adquirir la información (puntos y alarmas) y visualizarlaen un entorno web. La solución debe ser capaz de realizar el proceso de autenticación y carga de la configuración, así como de visualizar los diferentes despliegues, las últimas 5 alarmas emitidas por el sistema, el sumario de puntos, el sumario de alarmas y los detalles de un punto.

2.2. Exploración

En la fase exploración,los clientes plantean las historias de usuario que son de interés para la primera entrega del producto. Mientras el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se hacen pruebasa las diferentes tecnologías y seconstruye un prototipo para explorar las posibles arquitecturas del sistema. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología(16).

Para esta fase se decide utilizar 3 semanas, puesel cliente de visualización que se desea desarrollar no es un programa de gran extensión y el equipo de desarrollo tiene conocimiento acerca del trabajo con la mayoría de las herramientas que se utilizarán.

2.2.1. Actores del sistema

Tabla. 3 Autores del sistema.

Actores	**Descripción**
Usuario	Usuario que monitorea los datos

Tabla 1. Actoresdel sistema

2.2.2. Historias de usuario

Es la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, seanrequisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muydinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitadapara que los programadores puedan implementarla en unas semanas(17).

Los programadores estiman el esfuerzo asociado y las dependencias entre ellas. Para planificar el trabajo desde el punto de vista técnico, las HU son divididas en tareas, para las cuales también se realiza una estimación. Teniendo en cuenta el esfuerzo asociado a las HU y las prioridades del cliente se define una

versión que sea de valor y que tenga una duración de pocos meses.

Tabla. 4Autenticar Usuario.

Historia de Usuario	
Número: 1	**Nombre Historia de Usuario:** Autenticar Usuario
Actor: Usuario	**Iteración Asignada:** 1
Prioridad de Negocio: Alta	**Puntos Estimados:** 0.5
Nivel de Complejidad: Media	**Puntos Reales:** 0.5
Descripción: El cliente debe permitir autenticar usuarios del sistema SCADA Guardián del Alba. Los datos asociados son: usuario y contraseña.	
Observaciones: El usuario queda autenticado con los permisos necesarios.	

Tabla. 5Cargar Configuración.

Historia de Usuario	
Número: 2	**Nombre Historia de Usuario:** Cargar Configuración
Actor: Usuario	**Iteración Asignada:** 1
Prioridad de Negocio: Alta	**Puntos Estimados:** 0.5
Nivel de Complejidad: Media	**Puntos Reales:** 0.5
Descripción: El cliente debe permitir cargar la configuración asociada al usuario.	
Observaciones: La aplicación debe estar configurada.	

Tabla. 6Visualizar Despliegues.

Historia de Usuario	
Número: 3	**Nombre Historia de Usuario:** Visualizar Despliegues
Actor: Usuario	**Iteración Asignada:** 1
Prioridad de Negocio: Alta	**Puntos Estimados:** 1
Nivel de Complejidad: Alta	**Puntos Reales:**1

Descripción: El cliente debe permitir visualizar los despliegues asociados a la configuración del usuario
Observaciones:

Tabla. 7Visualizar Alarmas.

Historia de Usuario	
Número: 4	**Nombre Historia de Usuario:** Visualizar Alarmas
Actor: Usuario	**Iteración Asignada:** 1
Prioridad de Negocio: Alta	**Puntos Estimados:** 1
Nivel de Complejidad: Alta	**Puntos Reales:**1
Descripción: El cliente debe permitir visualizar las últimas 5 alarmas generadas por el Sistema SCADA Guardián del Alba.	
Observaciones:	

Tabla. 8Visualizar Sumario de Puntos.

Historia de Usuario	
Número: 5	**Nombre Historia de Usuario:** Visualizar Sumario de Puntos
Actor: Usuario	**Iteración Asignada:** 2
Prioridad de Negocio: Media	**Puntos Estimados:** 1
Nivel de Complejidad: Alta	**Puntos Reales:**1
Descripción: El cliente debe permitir visualizar el sumario de puntos del Sistema SCADA Guardián del Alba.	
Observaciones:	

Tabla. 9Visualizar Sumario de Alarmas.

Historia de Usuario	
Número: 6	**Nombre Historia de Usuario:** Visualizar Sumario de Alarmas
Actor: Usuario	**Iteración Asignada:** 2
Prioridad de Negocio: Media	**Puntos Estimados:** 1

Nivel de Complejidad: Alta	**Puntos Reales:**1
Descripción: El cliente debe permitir visualizar las alarmas generadas por el Sistema SCADA Guardián del Alba.	
Observaciones:	

Tabla. 10Visualizar Detalles de Puntos.

Historia de Usuario	
Número: 7	**Nombre Historia de Usuario:** Visualizar Detalles de Punto
Actor: Usuario	**Iteración Asignada:** 2
Prioridad de Negocio: Media	**Puntos Estimados:** 1
Nivel de Complejidad: Alta	**Puntos Reales:**1
Descripción: El cliente debe permitir visualizar los detalles de un punto del Sistema SCADA Guardián del Alba.	
Observaciones:	

2.2.3. Diseño de Casos de Pruebas

A diferencia delas metodologías tradicionales, donde la fase de pruebas, incluyendo la definición de las mismas, es usualmente realizada sobre el final del proyecto, o sobre el final del desarrollo de cada módulo; la metodología XP propone un modelo inverso, en el que, lo primero que se escribe son las pruebas que el sistema debe pasar(18).

Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios definidas por el cliente. Las pruebas de aceptación son consideradas como pruebas de caja negra. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos(18).

Las pruebas de aceptación son creadas a partir de las HU. Durante una iteración la HU seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. El cliente o usuario especifica los aspectos a probar cuando una HU ha sido correctamente implementada. Una HU puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento.Una

historia de usuario no se puede considerar terminada hasta que no pase correctamente todas las pruebas de aceptación.

Para la realización de las pruebas a la herramienta se diseñaron 8casos de pruebas donde se ejecutan paso a paso cada una de las posibles entradas al sistema y se evalúa el resultado de las mismas.

Las tablas con los diseños de casos de pruebas se encuentran en el epígrafe 3.1 del capítulo 3 donde además se le incluyen los resultados de la ejecución de las mismas.

2.3. Planificación y entrega

La planificación es una fasede pocos días,en la que el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. El resultado de esta fase es un Plan de Entregas.

Típicamente, esta fase consiste en una o varias reuniones grupales de planificación, donde se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debe obtenerse en no más de tres meses.

2.3.1. Plan de Entrega

Una vez que el cliente culmina la elaboración de las HU, se comienza con la creación del Plan de Entregas. El mismo se hace con la intención de que los programadores obtengan una estimación de dichas historias en cuanto al nivel de detalle, o sea, para fijar el período de tiempo que se puede tardar en la implementación de cada una.

Tabla. 11Plan de Entrega.

Historias de usuario	Iteración 1 (7 /3/2012)	Iteración 2 (21/3/2012)

Autenticar Usuario Cargar Configuración Visualizar Despliegues Visualizar Alarmas	3	Terminado
Visualizar Sumario de Puntos Visualizar Sumario de Alarmas Visualizar Detalles de Punto	No empezado	3

Tabla. 12Estimación del esfuerzo por historia de usuarios.

Historias de usuario	Tiempo estimado (semana ideal de trabajo)	Iteración asignada	Tiempo real
Autenticar Usuario Cargar Configuración Visualizar Despliegues Visualizar Alarmas	3	1	3
Visualizar Sumario de Puntos Visualizar Sumario de Alarmas Visualizar Detalles de Punto	3	2	3

A pesar de que se ha realizado una planificación que cumple con todos los requerimientos, los riesgos son altos. Para el desarrollo de la solución de forma segura y eficientese tomaron las siguientes medidas.

- Mantener un control de versiones del desarrollo de la aplicación, que permita regresar a una versión anterior y funcional si ocurriese algún problema.

- Comprobación periódica que permita probar el funcionamiento correcto de la aplicación y evitar que algunas funcionalidades presenten errores, debido a la incorporación de otras,o por el constante cambio en el código.

2.4. Diseño del sistema

La metodología XP no requiere la descripción del sistema por medio de diagramas de clase utilizando notación UML, sino que se guía por técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). Esto no implica que no se utilicen los diagramas para obtener una mejor visión y comunicación entre el equipo de trabajo, siempre y cuando su complejidad no sea alta y defina información importante.

2.4.1. Tarjetas CRC

Las características más sobresalientes de las tarjetas CRC son su simpleza y ductilidad. Una tarjeta CRC no es más que una ficha de papel o cartón que representa a una entidad del sistema(19).

Estas tarjetas se utilizan para estructurar las clases y a su vez definir las responsabilidades sobre las mismas, así como la simulación de escenarios en el sistema.

Tabla. 13Modelo de Tarjeta CRC.

Clase	
Responsabilidades	Colaboradores

Tabla 2. Modelo de Tarjeta CRC.

Tarjetas CRC del sistema

El sistema a desarrollar dispone de diversos objetos, por lo que se definen las siguientes clases:

xstorm-common-websocket,xstorm-common-command, xstorm-login, xstorm-explorer,xstorm-screenview, xstorm-common-screen,xstorm-summary,xstorm-screen,xstorm-graphic,xstorm-detail-point, xstorm-alarms

A continuación se definen las tarjetas CRC del sistema:

Tabla. 14Tarjeta CRC para la clase xstorm-common-websocket.

xstorm-common-websocket	
Crear conexiones websocket	

Tabla. 15Tarjeta CRC para la clase xstorm-common-command.

xstorm-common-command	
Implementarel PatrónCommand	xstorm-common-websocket

Tabla. 16Tarjeta CRC para la clase xstorm-login.

xstorm-login	
Gestionar la autenticación	xstorm-common-command

Tabla. 17Tarjeta CRC para la clase xstorm-console.

xstorm-explorer	
Se encargar devisualizar el árbol del proyecto	

Tabla. 18Tarjeta CRC para la clase xstorm-screenview.

xstorm-screenview	
Se encarga de controlar los diferentes despliegues y sumarios que se visualicen	xstorm-screen xstorm-summary

Tabla. 19Tarjeta CRC para la clase xstorm-screen.

xstorm-common-screen	
Crea un interfaz para los diferentes despliegues y sumarios	

Tabla. 20Tarjeta CRC para la clase xstorm-summary.

xstorm-summary	
Controlar la visualización del sumario de puntos	xstorm-common-screen xstorm-common-command

Tabla. 21Tarjeta CRC para la clase xstorm-screen.

xstorm-screen	
Controlar la configuración yvisualización de un despliegue	xstorm-common-screen xstorm-common-command xstorm-common-graphic xstorm-detail-point

Tabla. 22Tarjeta CRC para la clase xstorm-common-graphic.

xstorm-common-graphic	
Controlar la visualización de un objeto gráfico	

Tabla. 23Tarjeta CRC para la clase xstorm-detail-point.

xstorm-detail-point	
Controlar la visualización de los detalles de un punto.	

Tabla. 24Tarjeta CRC para la clase xstorm-alarms.

xstorm-alarms	
Controlar la visualización de las últimas 5 alarmas.	xstorm-common-websocket

2.4.2. Diagrama de Despliegue

La metodología XP plantea que para un mejor entendimiento de las tareas, flujos y métodos de desarrollo de las funcionalidades, se pueden crear diagramas, siempre que su creación no implique mayor esfuerzo que la implementación del mismo. Siguiendo este principio, se elaboró el diagrama de despliegue que permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes en la aplicación. Un diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema, en términos de cómo se distribuye la funcionalidad entre nodos de cómputo(20).

El sistema implementado se obtiene desde un servidor Web Apache y se ejecuta en la PC Cliente que se conecta al servidor XStormServer y este a su vez se conecta con la capa de comunicación del SCADA Guardián del Alba.

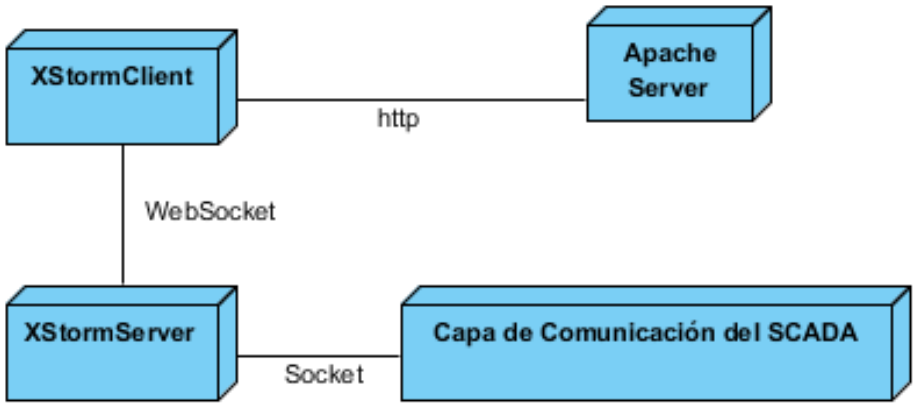


Figura. 4Diagrama de Despliegue.

2.4.3. Arquitectura del sistema

El diseño de la arquitectura de un sistema, es el proceso por el cual se define una solución para los requisitos técnicos y operacionales del mismo. Este proceso define qué componentes conforman el sistema, cómo se relacionan entre ellos, y cómo mediante su interacción llevan a cabo la funcionalidad especificada.

El diseño arquitectónico define la relación entre los elementos estructurales principales del software, los patrones de diseño que se pueden utilizar para lograr los requisitos que se han definido para el sistema, y las restricciones que afectan a la manera en que se pueden aplicar los patrones de diseño arquitectónicos(21).

Los patrones expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos; así como ayudan a especificar la estructura fundamental de una aplicación(22).

Se hará uso de una Arquitectura de Aplicaciones Javascript Escalable, definida por Nicholas Zakas en la conferencia “Scalable JavaScript ApplicationArchitecture”(23). Dicha arquitectura se divide en tres capas que tienen diferentes propósitos e interactúan entre sí, pero de manera independiente. A continuación se describen:

Biblioteca Javascript (YUI 3.x):

Brinda herramientas necesarias para un desarrollo rápido y robusto. Dentro de ellas se encuentra: la normalización de navegadores, parse y serialización de XML y JSON, manipulación de objetos, manipulación de DOM, comunicación Ajax y extensiones de código a bajo nivel.

Núcleo de la Aplicación:

Esta capa se encarga de manejar los diferentes módulos, así como de controlar la comunicación entre ellos, además de brindar extensiones para aumentar la capacidad de la aplicación.

Sandbox:

Brinda una interfaz de la cual heredan los diferentes módulos, teniendo como característica principal, que los módulos solo tienen acceso a los datos y funciones que están definidosen su ámbito (scope) y que no

conocen el resto de la arquitectura. Con lo que se garantiza la seguridad, se disminuyen las posibilidades de errores y se puede crear nuevos módulos si afectar el resto de la aplicación.

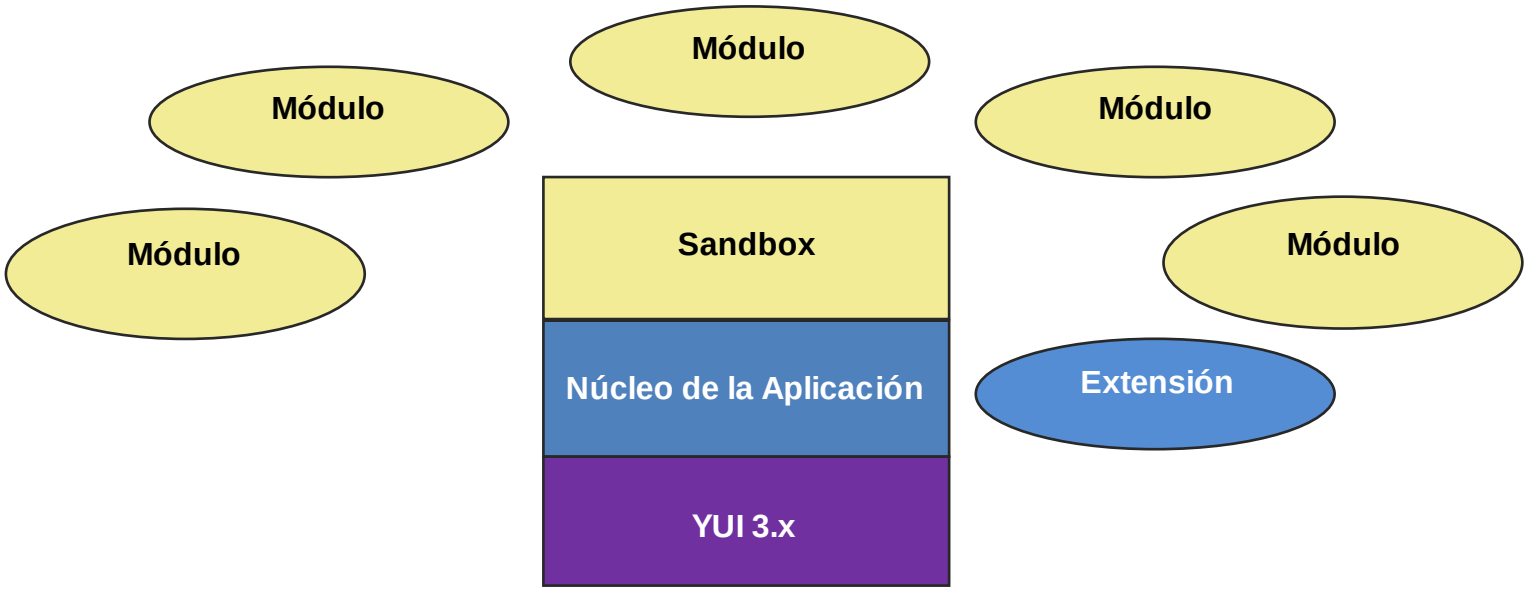


Figura. 5Arquitectura del sistema.

Principales módulos

A continuación de describen los principales módulos que posee la solución.

Módulologin: es el encargado de autenticar a los diferentes usuarios con el servidor de seguridad del sistema SCADA Guardián del ALBA.

Módulo titlebar: visualiza el nombre del usuario autenticado en la aplicación y brinda la opción de salir de la sesión.

Móduloexplorer: permite escoger entre los diferentes despliegues y sumarios configurados, el que se desea visualizar.

Módulo alarm: brinda la información de las últimas 5 alarmas generadas por el sistema SCADA Guardián del ALBA.

Módulo screen: es la representación gráfica de un despliegue del sistema SCADA Guardián del ALBA.

Módulo summary: visualiza información referente a la mayoría de los datos del sistema SCADA Guardián del ALBA, existe 3 tipo fundamentales: sumario de punto digitales, sumario de puntas analógico y sumario de alarmas.

Módulo screenview: encargado de manipular la visualización de los módulos screen y summary.

Módulo graphic: representa un objeto grafico específico, esto gráficos puede ser de varios tipos.

Módulo pointdetail: visualiza información más específica con respecto a un punto del sistema SCADA Guardián del ALBA.

Representación gráfica. (Ver Anexo 3)

2.4.4. Patrones de diseño

Los Patrones son soluciones comunes a problemas de diseño de software orientado a objetos y que además poseen ciertas características de efectividad para resolver ese problema. Son reusables ya que pueden ser aplicados en otros diseños o problemas(24).

Se hará uso de varios patrones de diseño, a continuación se mencionan los más importantes y se hace una breve descripción:

Patrón Instancia Única (Singleton):

Es uno de los patrones más básicos, pero útil, en JavaScript, y es probable que se utilice más que cualquier otro.Proporciona una forma de agrupar el código en una unidad lógica, que se puede acceder a través de una sola variable. Al asegurar que sólo existe un objeto único, se sabe que todo el código hace uso de los mismos recursos globales.

Se pueden utilizar como Namespace(Espacio de Nombre), lo que reduce el número de variables globales en sus páginas. Pueden ser utilizados para encapsularlas diferencias entre navegadores, a través de una técnica conocida como ramificación, lo que le permite utilizar las funciones más comunes de servicios

públicos, sin tener que preocuparse por el tipo de navegador. Lo más importante es que se puede utilizar para organizar el código de una manera consistente, lo que aumenta la legibilidad y facilidad de mantenimiento de suspáginas.

El uso de variables globales en aplicaciones web presenta un riesgo enorme, y un espacio de nombres creado con un producto único es una de las mejores maneras de eliminar las variables globales. Esto por sí solo hace que aumente el valor del patrón singleton, pero este patrón puede ser usado para muchos propósitos diferentes(24).

El uso de variables globalesen aplicaciones web presenta un riesgo enorme, y un espacio de nombres creado con un producto único es una de las mejores maneras de eliminar las variables globales. Esto por sí solo hace que aumente el valor del patrón singleton, pero este patrón puede ser usado para muchos propósitos diferentes(24).

En YUI 3.x se utiliza este patrón para la creación de las instancias globales de YUI y para la definición de diferentesNamespace (Y.namespac (´XStorm´);).

Patrón Fachada (Facade):

Este patrón hace dos cosas:simplifica la interfazde una clase, y separa esa clase del código del cliente que la utiliza. En JavaScript, las fachadas son a menudo el mejor amigo de un desarrollador. Siendo el principio básico detrás de casi todas lasbibliotecas de JavaScript. Dicho patrón puede hacer a los servicios públicos de las bibliotecas más fáciles de entender,mediante la creación de métodos de conveniencia, que permiten utilizar sistemas complejos de una manera másfácil y sencilla.

Proveer a los programadores de la capacidad de interactuar indirectamente con los subsistemas, lo cual es menos propenso a errores, que accederel subsistema de forma directa. Otras características importantes son: que simplifica las tareas comunes o repetitivas, permite añadir funciones convenientes a los objetos, usando métodos existente y combinación de ellos,simplifica las interfaces complejas, permite hacer la comprobación de errores detrás de las escenas, limpiar objetos de gran tamaño que ya no están en uso, y presentar las características de un objeto en una formamás fácil de usar.

Las fachadas no son estrictamente necesarias, se pueden realizar las mismas tareas sin su uso. Este es

un patrón de organización, que le permite modificar las interfaces de las clases y los objetos para que sea más conveniente para el programador, haciendoel código más manejable(24).

YUI 3.x utiliza este patrón con el fin de explotar las características antes mencionadas. Un ejemplo concreto es el tratamiento de eventos,quese hace de manera muy fácil y sencilla gracias a este patrón.

Patrón Observador (Observe):

En un entorno orientado a eventos, como son los navegadores web ,donde se busca constantemente la atención de un usuario, el patrón de observador, también conocido como el Patrón Publicador-Suscriptor, es una excelente herramienta para gestionar la relación entre las personas y sus puestos de trabajo, o más bien, los objetos, sus acciones, y su estado. En cuanto a JavaScript, este modelo le permite volver a observar el estado de un objetoen un programa y ser notificado cuando se cambia. En el patrón de observador, hay dos papeles: el observador y el observado(ver o ser visto) (24).

La aplicación desarrollada utiliza este patrón para trabajar con todos los eventos generados por el usuario y para la interrelación con los datos publicados desde el servidor.

Patrón Comando (Command):

Es una forma de encapsular la invocación de un método. Posee varias diferencias con respecto auna función normal, pues proporciona la capacidad de configurar y pasar de una llamada a un método, que luego puede ser ejecutado cada vez que se necesite. Tambiénpermite desacoplar el objeto, de la acción de invocar el objeto implementado, proporcionando así unenorme grado de flexibilidad en el intercambio entre clases concretas.Puede ser usado en situaciones diferentes, pero es muy útil en la creación de interfaces de usuario yser utilizado en lugar de una función de llamada, ya que permiteuna mayor modularidad en pasar la acción de un objeto a otro(24).

La aplicación desarrollada hace uso de este patrón específicamente en la interrelación entre las clases xstorm-common-websocket y otras como:xstorm-summary y xstorm-screen, para modular las acciones entre ellas.

Patrón Prototipo (Prototype):

Se puede desglosar en dos secciones principales, la sección de constructor y la sección del prototipo. Prototipos permite que las funciones y propiedades estén asociadas con los objetos. Sin embargo, en lugar de que cada instancia de un objeto obtenga una copia de todas las funciones y propiedades, lo que sucede es cada vez que se crea un objeto, utiliza el conjunto de funciones y propiedades que se definen para este tipo de objeto,dando como resultado un consumo de memoria inferior. En otras palabras, las funciones y propiedades se definen una vez por prototipo en lugar de una vez por objeto(25).

Este patrón lo utiliza JavaScript para aumentar las potencialidades del lenguaje.

2.4.5. Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. La metodología XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios(14). Para la implementación del sistema desarrollado se siguieron normas y estándares desarrollados, que se relacionan a continuación.

Definiciones generales

Las definiciones se realizan en inglés de manera descriptiva, evitando las abreviaturas y los nombres cortos.

Clases:

Las clases de la aplicación van a estar contenidas dentro del namespace“Y.XStorm”y van a comenzar con mayúscula al inicio de la palabra y en caso de estar conformada por palabras compuestas, la definición debe ser continua y cada palabra debe iniciar con mayúscula siguiendo el estilo determinado. El nombre del componente van comenzar por “xstorm-” y va a ser seguido por el nombre de la clase en minúscula.

Ejemplos:

```
Y.XStorm.Screen= Y.create('xstorm-screen', 'Y.Widget', [], {

});

Y.XStorm.Summary = Y.create('xstorm-summary, 'Y.Widget', [], {

});
```

Funciones:

Las funciones deben empezar con minúscula. En caso de estar conformadas por palabras compuestas, la definición debe ser continua y exceptuando la primera, cada palabra debe iniciar con mayúscula siguiendo el estilo determinado. En YUI 3.x se define como estándar de codificación que las funciones privada y protegidas debe empezar con guión bajo.

Ejemplos:

```
varinit = function(value){};

varstartComunication(){};
```

Función privada o protegida: var_destructor= function(){};

Variables:

Las variables comenzarán con la letra "x", aquellas que sean compuestas se escriben de manera seguida y con las primeras letras de cada palabra en mayúsculas sin incluir la primera.

Ejemplos:

```
varxid;

varxidComponent;
```

Sentencias simples

Cada línea debe contener una sola sentencia.

Ejemplo:

```
varxtabview = new Y.XStorm.TabView();

varxvalue =instance._calculate();
```

Sentencias compuestas

Deben estar indentadas a un nivel superior que el precedente. Todas las sentencias del tipo if, for, while, do... while deben tener llaves.

Ejemplos:

```
if (is_bool(val)){

varxid = instance.get('xid');

}

for ( var i =0; i < size; i++ ){
        if ( xuser == instance.get('xuser')[i] ){
            return true;
        }
    }
```

2.5. Desarrollo de Iteraciones.

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta etapa, generando al final de cada una, un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto.

En la primera iteración se puede intentar establecer una arquitectura del sistema, que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fuercen la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor del negocio). Al final de la última iteración el sistema estará listo para entrar en producción(14).

2.5.1. Implementación.

Las HU agrupadas en cada iteración se van implementando durante el transcurso de la iteración a la cual pertenecen. Al principio de cada iteración se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan se descomponen las HU en tareas de desarrollo, asignando a un grupo o una persona como responsable de su implementación. Estas tareas son para el uso estricto de los programadores, por lo que pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

A continuación quedan detalladas las tareas de desarrollo realizadas en cada una de las iteraciones.

Iteración 1

La primera iteración tendrá como objetivo darle cumplimiento a las HU 1,2,3,4que representan un mayor valor para el cliente, pues con las mismas se conformará la base de la estructura del negocio. Estas recogen funcionalidades de gran importancia para el proyecto, pues a través de ellas se definen aspectos que serán utilizados luego por las demás funcionalidades.

Tabla. 25Historias de usuarios planificadas para la primera iteración.

Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
Autenticar Usuario	0.5	0.5
Cargar Configuración	0.5	0.5
Visualizar Despliegues	1	1

Visualizar Alarmas	1	1

Tareas de las historias de usuarios desarrolladas en la primera iteración

Luego de relacionar las HU pertenecientes a esta iteración, se procede a la especificación de las principales tareas de desarrollo, que se realizaron para cumplir el propósito de la misma. (Anexo 1)

Iteración 2

La segunda iteración está centrada en desarrollar la HU 5, 6, 7 que son de gran importancia también para el cliente, además de que incluyen una mayor complejidad de desarrollo.(Anexo 2)

Tabla. 26Historias de usuarios planificadas para la segunda iteración.

Historia de usuario	Tiempo estimado (Semanas)	Tiempo real (Semanas)
Visualizar Sumario de Puntos	1	1
Visualizar Sumario de Alarmas	1	1
Visualizar Detalles de Punto	1	1

Tareas de las historias de usuarios desarrolladas en la segunda iteración

Las iteraciones de desarrollo sobre el sistema, permitieron que al finalizar se obtuviera un producto con todas las restricciones y características deseadas por el cliente.

Cabe destacar que el desarrollo de la solución se dividió en dos iteraciones para ordenar la implementación, realizándose pequeñas entregas al cliente al final de cada una de ellas, para recibir la

aceptación del mismo.

Consideraciones parciales

Conociendo como se lleva a cabo el flujo actual de eventos, se elaboró una propuesta del sistema. Se especificaron los usuarios que estarán relacionados con su utilización y funcionamiento. Se realizó una descripción de las HU precisando por el cliente la prioridad de cada una, definiendo así el orden en el que serán implementadas las mismas. Se cuenta con 7 HU que serán implementadas en dos iteraciones. Quedó elaborado el modelo necesario para llevar a cabo la implementación del sistema, mediante la descripción de los estándares de codificación y la arquitectura utilizada. Se plantearon y describieron las tareas de desarrollo, para dar cumplimiento a las funcionalidades abordadas por las HU. Se definió por medio del diagrama de despliegue la distribución física ,mediante la cual funcionará la aplicación y se diseñaron las pruebas de aceptación que determinan la confianza y seguridad de las funcionalidades del sistema para el cliente, la descripción de estas pruebas en conjunto con los resultados se encuentran en el siguiente capítulo.

Capítulo 3: Pruebas

Introducción

En el presente capítulo se especifican los resultados de la ejecución de las pruebas previamente diseñadas para probar las funcionalidades descritas.

3.1. Pruebas de Aceptación

La ejecución de las pruebas previamente diseñadas, permitió evaluar las funcionalidades de la herramienta antes de pasarla a un entorno real de explotación. Los resultados de las mismas se encuentran a continuación:

Tabla. 27Caso de prueba Autenticar Usuario correctamente.

Caso de prueba de Aceptación	
Número: 1	**Historia de Usuario:** 1
Nombre: Autenticar Usuario correctamente	
Descripción: El caso de prueba permite al usuario autenticarse en la aplicación.	
Condiciones de ejecución:	
Entradas/Pasos de ejecución: El usuario entra el usuario y la contraseña correctamente. -El sistema verifica los datos con el servidor de seguridad del Sistema SCADA Guardián del ALBA. -El sistema dispara el proceso de configuración.	
Resultado esperado: El sistema dispara el proceso de configuración.	
Evaluación de la prueba: Satisfactorio	

Tabla. 28Caso de prueba Autenticar Usuario incorrectamente.

Caso de prueba de Aceptación	
Número: 2	**Historia de Usuario:** 1
Nombre: Autenticar Usuario incorrectamente	
Descripción: El caso de prueba no permite al usuario autenticarse en la aplicación.	
Condiciones de ejecución:	

Entradas/Pasos de ejecución: El usuario entra el usuario y la contraseña incorrectamente. -El sistema verifica los datos con el servidor de seguridad del Sistema SCADA Guardián del ALBA. -El sistema visualiza el mensaje de error enviado por el servidor.
Resultado esperado: El sistema visualiza el mensaje de error enviado por el servidor.
Evaluación de la prueba: Satisfactorio

Tabla. 29Caso de prueba Cargar Configuración.

Caso de prueba de Aceptación	
Número: 3	**Historia de Usuario:** 2
Nombre: Cargar Configuración	
Descripción: El caso de prueba permite al sistema cargar la configuración.	
Condiciones de ejecución: El usuario debe de estar autenticado.	
Entradas/Pasos de ejecución: -El sistema pide la configuración al servidor. - El servidor envía la configuración. - El sistema se configurara.	
Resultado esperado: El sistema configurado.	
Evaluación de la prueba: Satisfactorio	

Tabla. 30Caso de prueba Visualizar de Despliegue.

Caso de prueba de Aceptación	
Número: 4	**Historia de Usuario:** 3
Nombre: Visualizar de Despliegue	
Descripción: El caso de prueba permite al usuario visualizar diferentes despliegues.	
Condiciones de ejecución: El usuario debe de estar autenticado. Deba existir Despliegues.	
Entradas/Pasos de ejecución: El usuario escoge un despliegue. -El sistema debe visualizar este desplieguey para de visualizar los demás despliegues.	
Resultado esperado: El sistema visualiza el despliegue escogido.	
Evaluación de la prueba: Satisfactorio	

Tabla. 31Caso de prueba Visualizar Alarmas.

Caso de prueba de Aceptación	
Número: 5	**Historia de Usuario:** 4
Nombre: Visualizar Alarmas	
Descripción: El caso de prueba permite al usuario visualizar las últimas 5 alarmas.	
Condiciones de ejecución: El usuario debe de estar autenticado.	
Entradas/Pasos de ejecución: -El sistema debe visualizar las 5 últimas alarmas.	
Resultado esperado: El sistema debe visualizar las 5 últimas alarmas.	
Evaluación de la prueba: Satisfactorio	

Tabla. 32Caso de prueba Visualizar Sumario de Alarmas.

Caso de prueba de Aceptación	
Número: 6	**Historia de Usuario:** 1
Nombre: Visualizar Sumario de Alarmas	
Descripción: El caso de prueba permite al usuario visualizar el sumario de alarmas.	
Condiciones de ejecución: El usuario debe de estar autenticado.	
Entradas/Pasos de ejecución: El usuario escoge visualizar el sumario de alarmas. -El sistema debe visualizar el sumario de alarmas y para de visualizar los despliegues.	
Resultado esperado: El sistema visualiza el sumario de alarmas.	
Evaluación de la prueba: Satisfactorio	

Tabla. 33Caso de prueba Visualizar Sumario de Puntos.

Caso de prueba de Aceptación	
Número: 7	**Historia de Usuario:** 1
Nombre: Visualizar Sumario de Puntos	
Descripción: El caso de prueba permite al usuario visualizar el sumario de puntos.	
Condiciones de ejecución: El usuario debe de estar autenticado.	
Entradas/Pasos de ejecución: El usuario escoge visualizar el sumario de puntos. -El sistema debe visualizar el sumario de puntos y para de visualizar los despliegues.	
Resultado esperado: El sistema visualiza el sumario de puntos.	

Evaluación de la prueba: Satisfactorio

Tabla. 34Caso de prueba Visualizar Detalles de Punto.

Caso de prueba de Aceptación	
Número: 8	**Historia de Usuario:** 1
Nombre: Visualizar Detalles de Punto	
Descripción: El caso de prueba permite al usuario visualizar los detalles de punto.	
Condiciones de ejecución: El usuario debe de estar autenticado. Deba existir objetos gráficos.	
Entradas/Pasos de ejecución: El usuario escoge visualizar los detalles de un punto en específico. -El sistema debe visualizar los detalles del punto escogido y continuara visualizando los despliegues.	
Resultado esperado: El sistema visualiza los detalles del punto.	
Evaluación de la prueba: Satisfactorio	

Consideraciones Parciales

Se ejecutaron las pruebas de aceptación, que permitieron demostrar que el sistema se encuentra listo y cumple con las exigencias de los clientes.

Conclusiones

Una vez finalizada la investigación se puede concluir que:

- Se realizó un estudio y análisis de las herramientas y tecnologías, utilizadas para la visualización de la información de los sistemas SCADA sobre plataforma web, lo que permitió un desarrollo más rápido y eficaz de la solución.
- Se logró la generación de los artefactos propuestos por la metodología seleccionada para el desarrollo de la solución, con lo cual quedaron definidos los requerimientos exigidos por el cliente.
- Fue probado el sistema, lo que demostró el cumplimiento de las exigencias del cliente.
- Quedó implementada la herramienta informática propuesta, permitiendo que se visualice en navegadores web la información del sistema SCADA Guardián del ALBA.

Recomendaciones

Concluido el desarrollo de este trabajo se recomienda:

- Desarrollar una biblioteca de componentes gráficos para aumentar las posibilidades de integración con los diferentes componentes del sistema SCADA Guardián del ALBA.
- Personalizar la interfaz para una mejor experiencia de usuario en Smartphone (Teléfonos Inteligentes) y Tables (Tabletas).
- Integrar otros módulos y aplicaciones relacionadas con el Sistema SCADA Guardián del ALBA en esta solución.

ReferenciasBibliográficas

1. **Sehara, Yossel Luis.** *Implementación de un modelo para la configuración de un sistema SCADA.* Cuidad de la Habana : s.n., 2008.
2. **Bernardo Zaragoza Hijuelos, Raudi Agdel Bacallao.** *Implementación de un módulo de generación de reportes para sistemas de supervisión, control y adquisición de datos.* Ciudad de La Habana : s.n., julio, 2008. .
3. **Feria, Osmany Pérez.** *Estructura Base del Visualizador del SCADA Guardián del Alba.* Ciudad de la Habana : s.n., Abril, 2011.
4. **Garrett, Jesse James.** Adaptivepath. [En línea] Febrero de 2005. http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications.
5. **Saint-Andre, Peter, Smith, Kevin y Troncon, Remko.** *XMPP: The Definitive Guide.* San Peterburgo : O`Reilly Media, 2009.
6. **W3C.** W3C. [En línea] Octubre de 2011. http://www.w3.org/TR/eventsource/.
7. **-.** W3C. [En línea] Diciembre de 2011. http://www.w3.org/TR/websockets/.
8. **Pésez, Javier Eguíluz.** *Intrudución a XHTML.*
9. [En línea] http://mlw.io/guia-html5/ .
10. **Pérez, Javier Eguíluz.** *Introducción a CSS.*
11. **W3C.** W3C. [En línea] Agosto de 2011. http://www.w3.org/TR/SVG11/.
12. **Stefanov, Stoyan.** *JavaScript Patterns.*
13. **Yahoo.** YUI Library. [En línea] http://yuilibrary.com/.
14. **Letelier, Patricio.** *Metodologías Ágiles en el Desarrollo de Software.* 2008.
15. *Métodologías Ágiles en el Desarrollo de Software.* **José H. Canós, Patricio Letelier y Mª Carmen Penadés.** Universidad Politécnica de Valencia : s.n.
16. **Gómez Argüello, Wilson Javier.** Metodología de desarrollo de software un enfoque práctico y global versión 1.0.11. [En línea] http://www.otcolombia.com/documentos/mds360-1.0.11-beta.pdf..
17. **Beck, K..** *Extreme Programming Explained. Embrace Change.* s.l. : Pearson Education, 1999.
18. **Joskowicz, José.** Reglas y Prácticas en eXtreme Programming. [En línea] 2008. http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf.
19. **Casas, Sandra y Reinaga, Héctor.** Identificación y Modelado de Aspectos Tempranos dirigido por Tarjetas de Responsabilidades y Colaboraciones. [En línea] 2008. http://www.oocities.org/espanol/profeprog2/INVPAPER25.pdf.
20. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** El Proceso Unificado de Desarrollo

de Software. [En línea] 2000. http://bibliodoc.uci.cu/pdf/8478290362.pdf.
21. **Shaw, M. y Garlan, D.** *"Software Architecture".* New York : Prentic-Hall, 1996.
22. **Pressman, Roger S.** *"Ingeniería de Software. Un enfoque práctico." 5ta Edición (traducción de la edición original en Inglés"Software Engineering. A practical approach").* Madrid : Mac Graw Hill, 2001.
23. **Zakas, Nicholas.** Scalable JavaScript Application Architecture. [En línea] Septiembre de 2009. http://www.yuiblog.com/blog/2009/09/17/video-bayjax-sept-09/.
24. **Diaz, Dustin y Harmes, Ross.** *Pro JavaScript Design Patterns.* New York : apress, 2008.
25. **Wahlin, Dan.** The Prototype Pattern - Techniques, Strategies and Patterns for Structuring JavaScript Code. [En línea] 2011. http://weblogs.asp.net/dwahlin/archive/2011/08/01/techniques-strategies-and-patterns-for-structuring-javascript-code-the-prototype-pattern.aspx.

Glosario de Términos

ERP: Los sistemas de Planificación de Recursos Empresariales, o ERP (Enterprise ResourcePlanning) son Sistemas de Información Gerenciales que integran y manejan muchos de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía en la producción de bienes o servicios.

RTU: Unidad Terminal Remota es un dispositivo electrónico controlado por microprocesador que interconecta los objetos del mundo físico a un sistema de control distribuido o SCADAmediante la transmisión de datos de telemetría para el sistema, y mediante el uso de mensajes desde el sistema de control para controlar objetos conectados.

PLC: Los Controladores Lógicos Programables o PLC (ProgramableLogicController) son dispositivos electrónicos muy usados en automatización industrial.

XML: Lenguaje de Marcas Extencible (eXtensible Markup Language) es un lenguaje de marcas desarrollado por el World Wide Web Consortium(W3C).

JSON:acrónimo de Notación de Objetos Javascript (Javascript ObjectNotation), es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

XMLHttpRequest (XHR): también referida como XMLHTTP (Extensible MarkupLanguage / Hypertext Transfer Protocol), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores Web.

HTTP: Protocolo de Transferencia de Hipertexto (Hypertext Transfer Protocol) es el protocolo usado en cada transacción de la World Wide Web.

TCP/IP:es un modelo de descripción de protocolos de red, que describe un conjunto de guías generales de diseño e implementación de protocolos de red específicos, para permitir que un equipo pueda comunicarse. Además provee conectividad de extremo a extremo especificando como los datos deberían ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario.

Cross Domain: Es un mecanismo de seguridad de las comunicaciones en navegadores actuales. Evitan que un script (XMLHttpRequest de AJAX) o una aplicación (Flash, Silverlight) de una página web puedan acceder a un servidor web diferente del que residen.

Anexo 1

Tareas de desarrollo de la primera iteración

Crear extensión de comunicación

Tarea	
Número de la tarea:1	**Número de HU:** 1,2,3,4
Nombre de la tarea: Crear extensión de comunicación	
Tipo de tarea: Diseño	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Esta tarea crea la extensión de la capa de la aplicación que permite obtener datos de servidor a través de protocolo websocket.	

Autenticar Usuarios

Tarea	
Número de la tarea:2	**Número de HU:** 1
Nombre de la tarea: Autenticar Usuarios	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Con esta tarea se permite el acceso a los usuarios del Sistema SCADA Guardián del ALBA.	

Crear interfaz de la aplicación

Tarea	
Número de la tarea:3	**Número de HU:** 3,4
Nombre de la tarea: Crear interfaz de la aplicación	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Con esta tarea se crea una interfaz donde posteriormente se insertan	

los diferentes componentes de la aplicación.

Cargar Configuración

Tarea	
Número de la tarea:4	**Número de HU:** 2
Nombre de la tarea: Cargar la configuración	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Con esta tarea permite inicializar la aplicación con la configuración adecuada.	

Visualizar Despliegues

Tarea	
Número de la tarea:5	**Número de HU:** 3
Nombre de la tarea: Visualizar Despliegues	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Con esta tarea permite visualizar los despliegues.	

Visualizar Alarmas

Tarea	
Número de la tarea:6	**Número de HU:** 4
Nombre de la tarea: Visualizar Alarmas	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Con esta tarea permite visualizar las alarmas.	

Anexo 2

Tareas de desarrollo de la segunda iteración

Crea módulo de Sumario

Tarea	
Número de la tarea:7	**Número de HU:** 5,6
Nombre de la tarea: Crear módulos Sumario	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Con esta tarea permite crear un módulo para la visualización de sumarios.	

Visualizar Sumario de Alarmas

Tarea	
Número de la tarea:8	**Número de HU:** 5
Nombre de la tarea: Visualizar Sumario de Alarmas	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Con esta tarea permite visualizar el sumario de alarmas.	

Visualizar Sumario de Puntos

Tarea	
Número de la tarea:9	**Número de HU:** 6
Nombre de la tarea: Visualizar Sumario de Puntos	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	

Descripción: Con esta tarea permite visualizar el sumario de Puntos.

Visualizar Detalles de Punto

Tarea	
Número de la tarea:10	**Número de HU:** 7
Nombre de la tarea: Visualizar Detalles de Punto	
Tipo de tarea: Desarrollo	**Puntos estimados**:
Programador(es) responsable(s): Yuri VictorMunayev	
Descripción: Con esta tarea permite visualizar el detalle de un Punto.	

Anexo 3

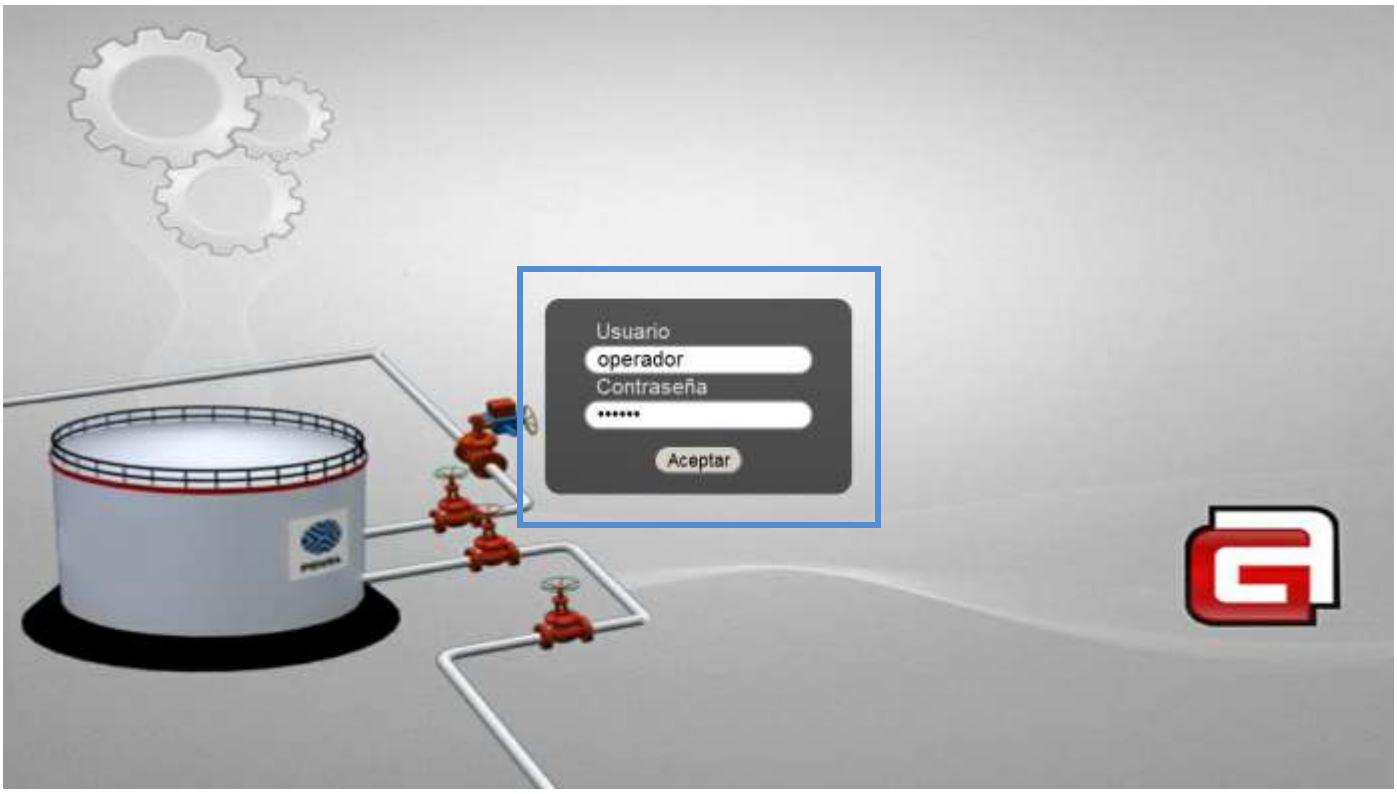


Figura. 5 Módulo login.

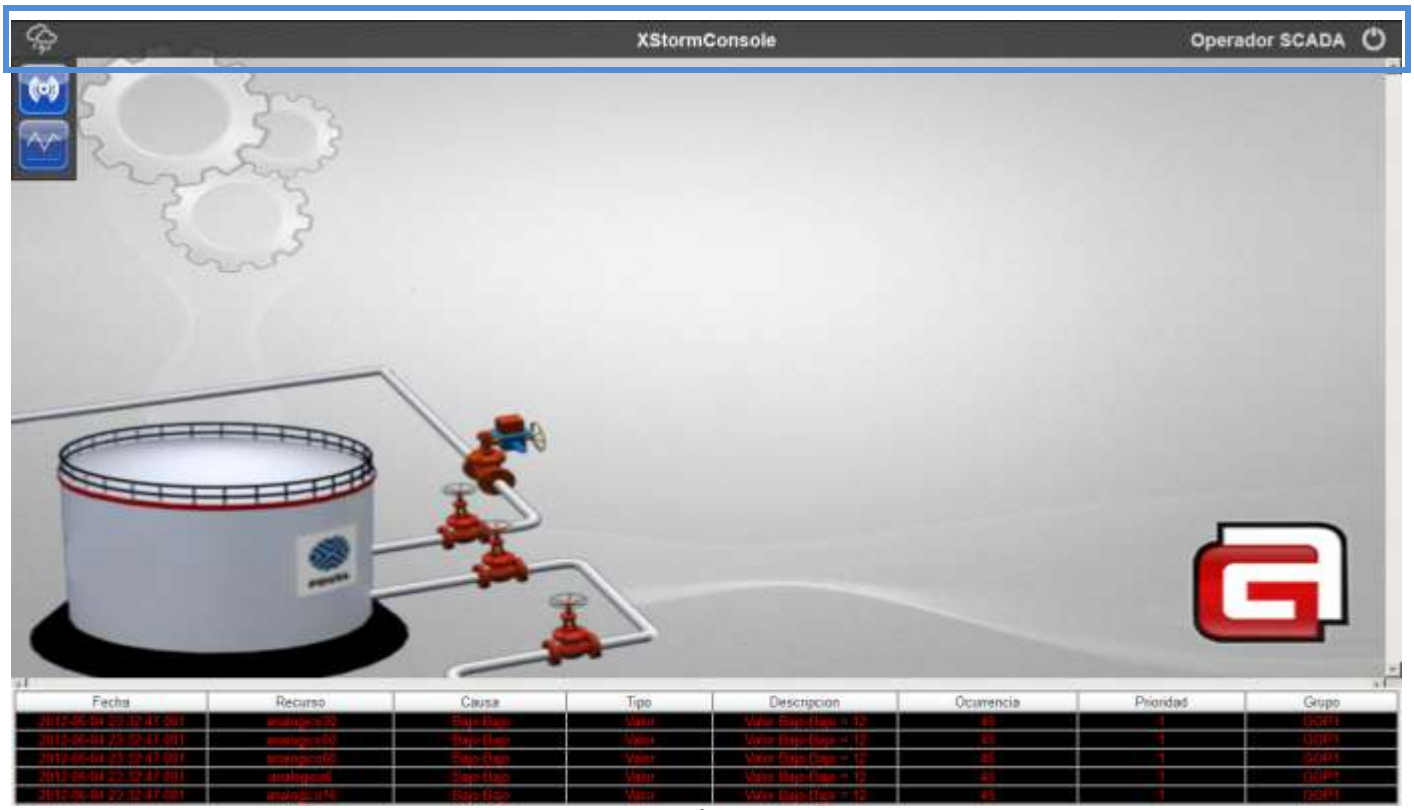
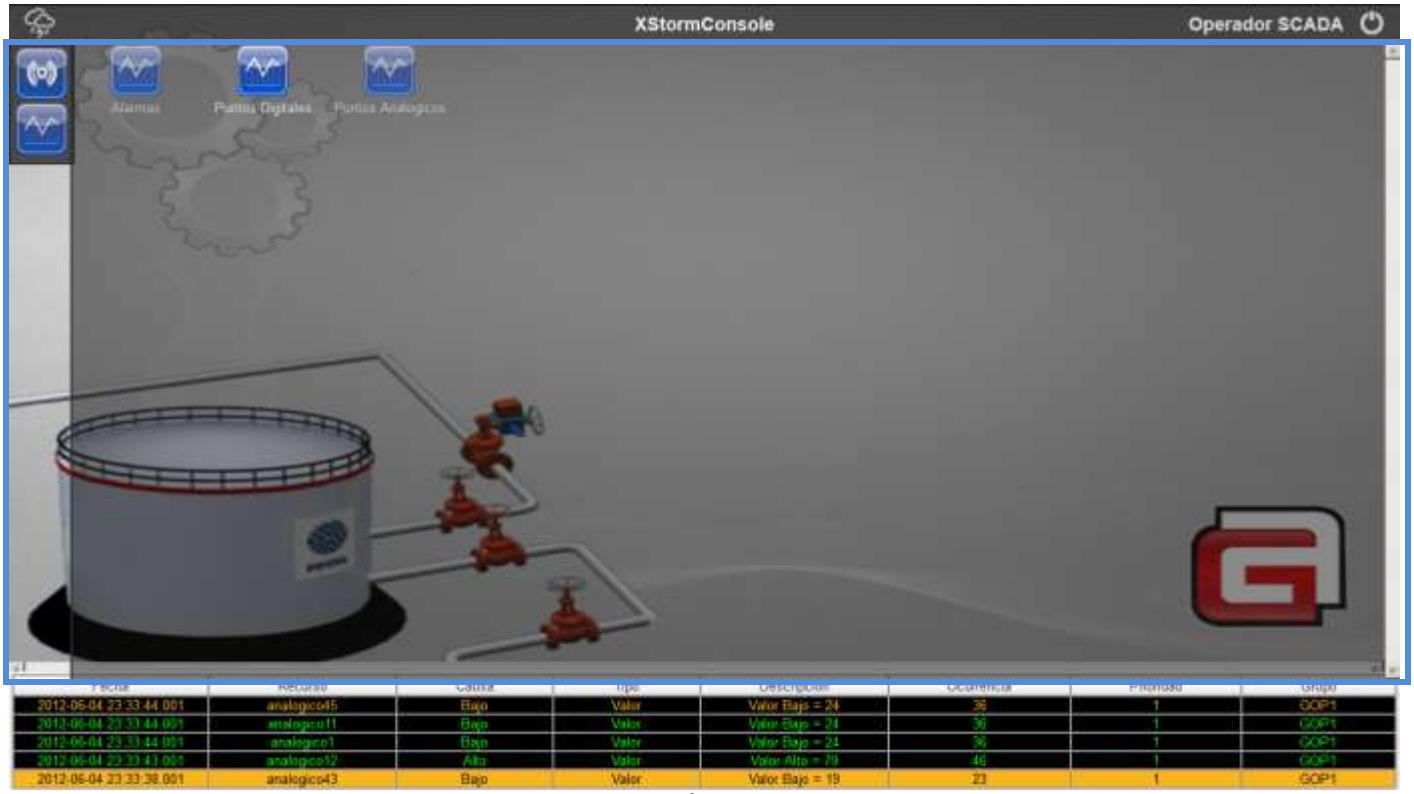


Figura. 6 Módulo titlebar.



Fecha	Recurso	Causa	Tipo	Descripción	Ocurrencia	Prioridad	Grupo
2012-06-04 23:33:44.001	analogico45	Bajo	Valor	Valor Bajo = 24	36	1	GOP1
2012-06-04 23:33:44.001	analogico11	Bajo	Valor	Valor Bajo = 24	36	1	GOP1
2012-06-04 23:33:44.001	analogico1	Bajo	Valor	Valor Bajo = 24	36	1	GOP1
2012-06-04 23:33:43.001	analogico12	Alto	Valor	Valor Alto = 79	46	1	GOP1
2012-06-04 23:33:38.001	analogico43	Bajo	Valor	Valor Bajo = 19	23	1	GOP1

Figura. 7 Módulo explorer.

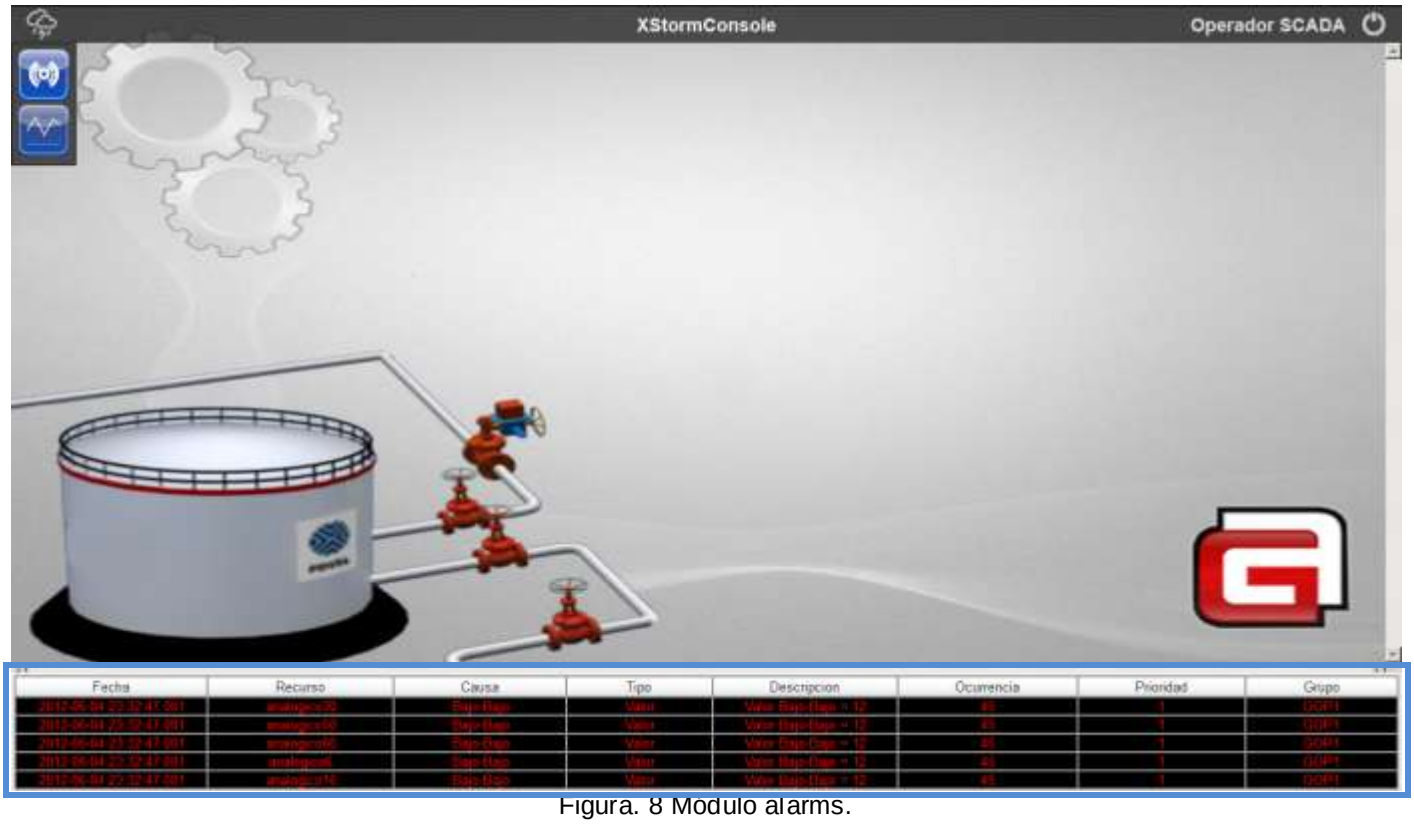


Figura. 8 Modulo alarms.

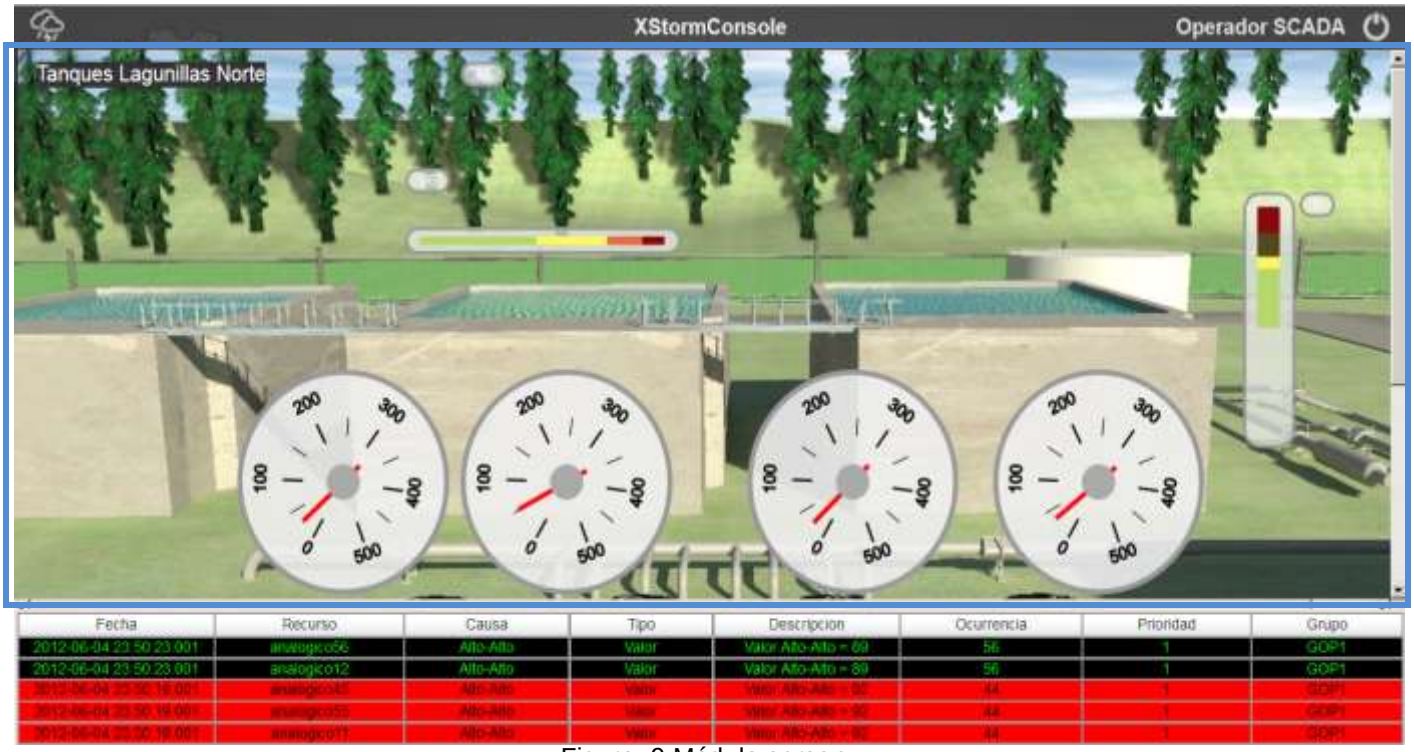


Figura. 9 Módulo screen.

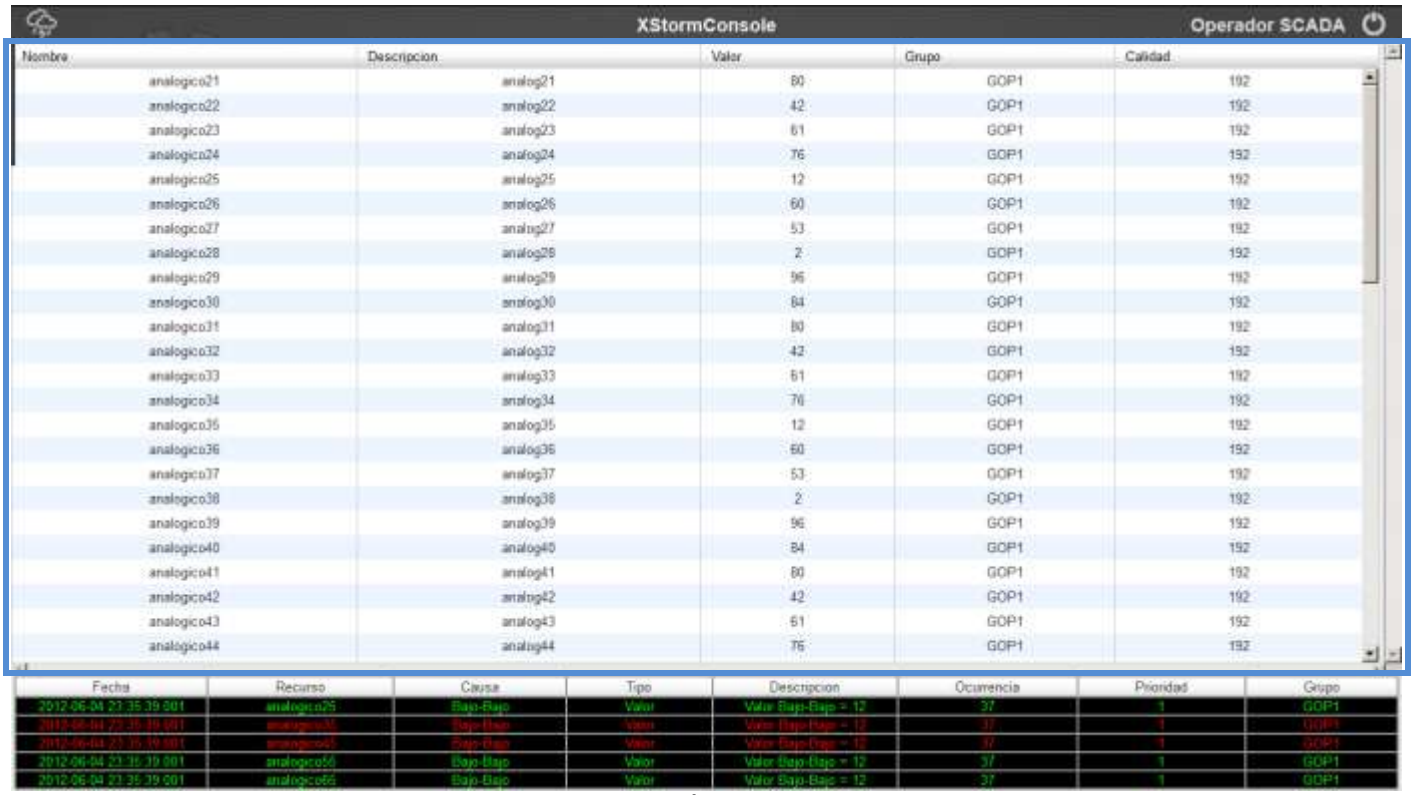


Figura. 10 Módulo summary.

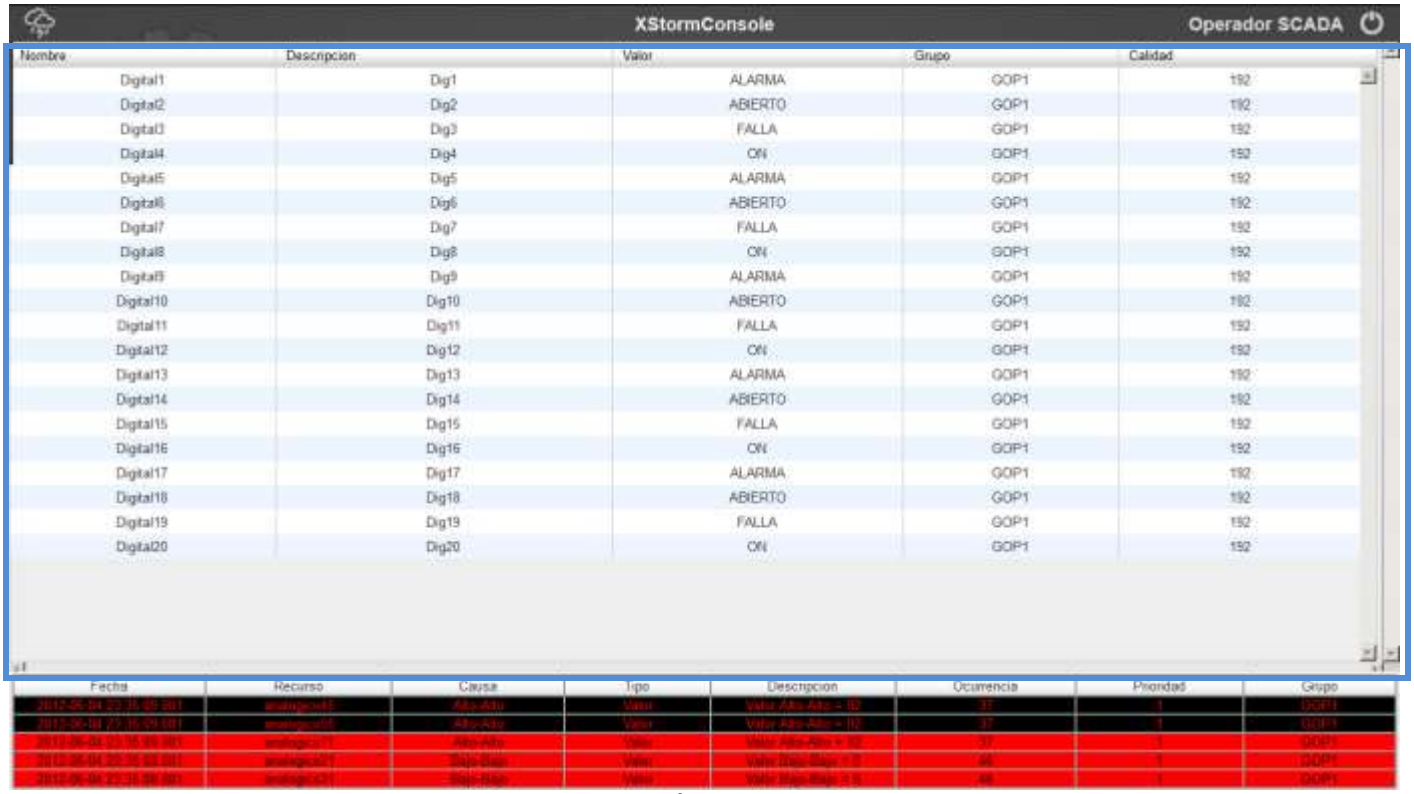


Figura. 11 Módulo screenview.

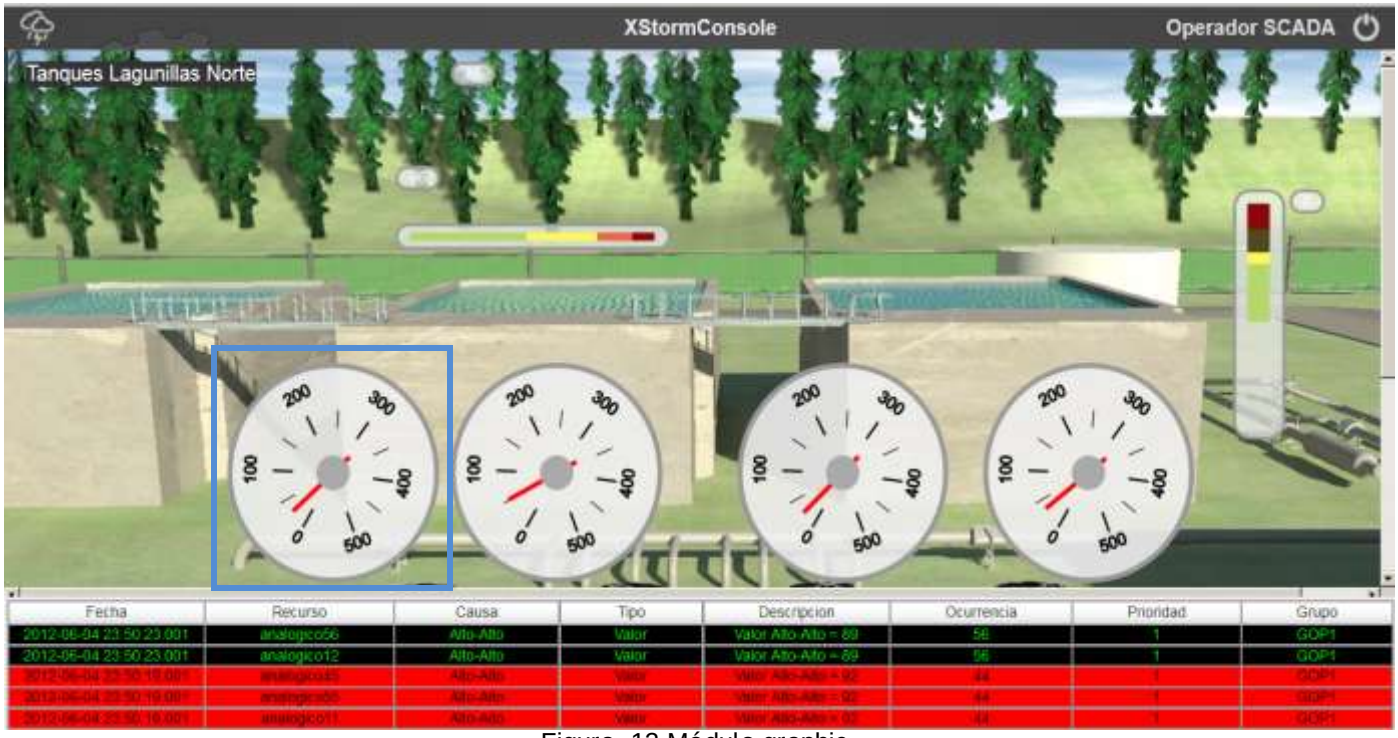


Figura. 12 Módulo graphic.

Figura. 13 Módulo pointdetail.