

Cliente HMI del SCADA Guardián del Alba para Dispositivos Móviles

Ing. José Enrique Díaz Ramos¹;

Ing. Rita Milena Hernández Díaz²

1. Universidad de Matanzas – Sede “Camilo Cienfuegos”, Vía Blanca Km.3, Matanzas, Cuba. enrique.diaz@umcc.cu

2. 1. Universidad de Matanzas – Sede “Camilo Cienfuegos”, Vía Blanca Km.3, Matanzas, Cuba. rita.hernandez@umcc.cu



Monografías



Resumen

El presente trabajo surge de la necesidad de acceso constante a la información generada por el Sistema SCADA (System Control and Data Adquisition) Guardián del Alba por parte de los usuarios implicados en la producción. Teniendo en cuenta que los dispositivos móviles pueden contribuir a aumentar la accesibilidad a la información, esta investigación se traza como objetivo desarrollar un cliente HMI(Human Machine Interface) que permita visualizar en dispositivos móviles la información supervisada por el Sistema SCADA Guardián del Alba. Para el cumplimiento del mismo se refleja una investigación acerca de los principales aspectos y estándares a tener en cuenta a la hora de diseñar clientes HMI. Además se describe la etapa de desarrollo de la aplicación, especificando los artefactos y funcionalidades fundamentales para darle cumplimiento al objetivo inicial.

Palabras claves: SCADA;HMI



Introducción

En la actualidad se han desarrollado sistemas para la supervisión y control de los procesos automatizados de las grandes industrias, estos sistemas son conocidos generalmente como SCADA. En la Universidad de las Ciencias Informáticas se desarrolla un sistema de este tipo para las refinerías de petróleo de PDVSA (Petróleos de Venezuela S.A.) conocido como Guardián del Alba.

Uno de los principales objetivos de este tipo de software es garantizar el acceso de los clientes a la información siempre que sea necesaria, para posibilitar un mayor control y ayudar a la toma de decisiones. La implantación exitosa del sistema en las diferentes áreas operacionales posibilita generar información valiosa sobre el comportamiento del proceso, el balance de la producción, la operatividad de los dispositivos de campo, el estado de las redes de comunicación entre otras informaciones de utilidad para operadores, mantenedores del sistema, ingenieros, supervisores y gerentes. Basado en lo anteriormente expuesto se han identificado una serie de requerimientos o necesidades que atentan contra la buena supervisión de los procesos controlados por el sistema.

Los procesos en la industria se monitorizan a través de consolas de trabajo –monitores, pantallas, paneles, etc.- donde los operadores se mantienen durante turnos de varias horas observando el proceso que supervisan, lo cual requiere la presencia permanente de una persona en el ambiente conectado a la red de servidores SCADA. En determinadas situaciones los usuarios se encuentran en lugares donde no tienen acceso a las salas de control donde se supervisa el proceso, lo que ocasiona falta de accesibilidad a la información, ejemplo de estas situaciones son: los mantenedores u operadores en planta deben llamar a las salas de control para conocer la medición que reporta cierta variable en el SCADA y así comprobar los ajustes realizados en campo, los gerentes o funcionarios que necesiten la información en todo momento estarían privados de la misma si estuvieran fuera de la sala de control, lo cual repercute en la correcta toma de decisiones críticas, esto ocasiona un ineficaz aprovechamiento de la información en la industria más las consiguientes pérdidas de tiempo y dinero.



Las características de los dispositivos móviles hacen de los mismos una solución viable a los problemas de accesibilidad de la información en el SCADA Guardián del Alba, ya que permitirían un aumento significativo en este aspecto, debido a sus grandes posibilidades de conectividad. De acuerdo a esto se identifica la posibilidad de desarrollar un cliente HMI que permita visualizar la información en dispositivos móviles independiente del sistema operativo, con garantías de alto rendimiento, integridad y fiabilidad de la información.

En un trabajo que precedió a esta investigación se propuso un cliente web para el HMI del SCADA que pretendía visualizar los procesos industriales del GALBA en dispositivos portátiles. La palabra portable no se define en el marco de dicha investigación aparentemente se refiere a otro tipo de dispositivos -laptops-, y a juzgar según los resultados presentados no se despliega la solución sobre dispositivos móviles, ya que la información que se presenta no tiene ningún tratamiento específico para ser visualizada en este tipo de dispositivos.

Por su parte el HMI de la versión Miranda R2, desarrollada en Java no es compatible para instalar en dispositivos móviles con los sistemas operativos Android, iOS, BlackBerry o Windows Phone debido a su dependencia con la máquina virtual de java Standart Edition(SE). Lo mismo ocurre con el HMI de la versión Zamora R1, actualmente en desarrollo con el framework Qt4.

Dada la situación problemática previamente expuesta se define como **objeto de estudio**: El proceso de visualización realizado por el Sistema SCADA Guardián del Alba. Definiendo como **objetivo general** desarrollar un cliente HMI que permita visualizar en dispositivos móviles, la información supervisada por el Sistema SCADA Guardián del Alba.

Desarrollo

Al planificar el desarrollo de esta aplicación se tienen que tomar en cuenta varios puntos esenciales, el primero lograr la conexión con el servidor del SCADA que se encontraba ya en funcionamiento, otro factor importante fue que el cliente específico entre los requisitos que la aplicación debía ser multiplataforma, siendo esencial que funcionara al menos en los sistemas operativos móviles IOS y Android. Para darle respuesta al primer punto se tiene



que el servidor del Guardián del Alba utiliza como protocolo de comunicación QtWebsocket, una implementación del protocolo Websocket realizada en C++, por lo tanto, se debe usar este protocolo en la implementación de la aplicación. Para darle respuesta al segundo punto, el ser multiplataforma se tuvieron en cuenta varias soluciones, la primera el desarrollar una aplicación nativa para cada sistema operativo. Esta solución se desechó en primera instancia ser poco eficiente pues se tiene que realizar el mismo trabajo 2 o más veces, quedándose esta solución como última opción. Otra solución fue el hacer una página web, esta solución cumplía con los requisitos de ser multiplataforma y de solo tener que desarrollar la aplicación una sola vez, pero el protocolo websocket no está estandarizado todavía y en algunos navegadores existirían problemas de comunicación con el servidor, esta opción fue desechada también. Se encontraron entonces 2 tecnologías que permitían desarrollar aplicaciones multiplataforma y que el producto desarrollado era una aplicación nativa de cada sistema operativa, la primera PhoneGap, esta permite desarrollar una aplicación en HTML5, posteriormente el desarrollador puede escoger entre una compilación local o subir su código a la página de PhoneGap y la compañía compila en los sistemas operativos de tu elección y envía la aplicación al desarrollador gratuitamente. El problema con esta tecnología es que la aplicación aunque parece nativa en realidad se ejecuta sobre el navegador del teléfono, y se repite el problema de las aplicaciones web. La segunda tecnología de este tipo estudiada fue QtQuick, esta es una aplicación multiplataforma y marco de interfaz de usuario para los desarrolladores que utilizan C++ o QML, CSS y JavaScript como lenguaje, teniendo a Qt Creator como el entorno de desarrollo que lo soporta. Además es una colección de tecnologías diseñadas para ayudar a los desarrolladores a crear interfaces de usuario modernas, fluidas e intuitivas que son muy usadas en teléfonos móviles, tablets y otros dispositivos portables. Esta permite fácilmente la integración con el protocolo de comunicación seleccionado.

Al escoger dicha tecnología se definen los lenguajes que serán utilizados en el desarrollo de la aplicación, destacando entre ellos el QML. Este es un lenguaje declarativo utilizado para el desarrollo de interfaces de usuario modernas e intuitivas, principalmente para aplicaciones móviles, donde la entrada táctil, las animaciones fluidas y una buena



experiencia de usuario son cruciales. Además define y permite a los desarrolladores de aplicaciones ampliar el lenguaje QML con tipos de datos personalizados e integrar código QML con JavaScript y C++. Se debe tener en cuenta que el módulo de QtQML proporciona solo el lenguaje y la infraestructura para QML, y no un lienzo visual o motor de renderizado para la creación de interfaces de usuario. Los mismos se brindan por el módulo QtQuick, que proporciona diversos componentes visuales, un framework para animación y mucho más para la construcción de interfaces de usuario con QML.

Después de darle solución a estos problemas se procede a definir las principales funcionalidades del software de acuerdo a los requisitos del cliente. Se define que se tendrán como funcionalidades principales la visualización de los datos de las alarmas, las señales analógicas y digitales, así como los eventos del sistema, los dispositivos y los subcanales. Además se deberán mostrar detalles de cada uno de ellos y específicamente en el caso de las señales se debe mostrar una gráfica con los valores en tiempo real. Otro requisito importante es poder filtrar los valores por diferentes parámetros. Por cuestiones de seguridad se determina también no permitir controlar, al menos en esta primera fase del proyecto, los procesos mediante la aplicación pues es muy común que se pierda el teléfono móvil y cualquier persona que se lo encuentre pudiera hipotéticamente controlar todo el proceso de la refinería.

Posteriormente se realiza un diseño de las interfaces basado en los requisitos previamente expuestos, el equipo define que se tendrán 5 interfaces que mostrarán los principales tipos de datos. Cada una de estas interfaces tendrá 3 columnas que desplegaran los datos esenciales para los trabajadores y permitirá al presionar cualquier elemento acceder a los detalles del mismo. En el caso de las señales específicamente esta interfaz contendrá tanto las analógicas como las digitales y se podrá cambiar entre ellas con un elemento deslizante situado en la parte superior de la interfaz. Los detalles, al igual que los elementos principales podrán variar acorde a las especificaciones que realice el cliente. El diseño de estas interfaces estuvo basado en varios estándares internacionales de automática, así como estándares propios de la entidad PDVSA.



Una vez terminado el diseño se procedió al proceso de desarrollo de la aplicación, el mismo duró aproximadamente 5 meses y tras pasar la fase de pruebas se aplicó obteniéndose buenos resultados.

Conclusiones

En el presente trabajo se propuso como objetivo principal desarrollar un cliente HMI que permita visualizar en dispositivos móviles, la información supervisada por el Sistema SCADA Guardián del Alba. Durante el desarrollo del mismo se le dio cumplimiento al objetivo trazado obteniendo lo siguientes resultados:

- Se logró un mecanismo para la visualización de la información supervisada por el SCADA GALBA en dispositivos móviles.
- Se validó el sistema mediante una serie de pruebas lo cual permitió verificar el cumplimiento de los requisitos establecidos por el cliente.



Referencias Bibliográficas

1. Drop Ingeniería S.L. SCADA. *Drop Ingeniería S.L.* [En línea] [Citado el: 5 de diciembre de 2013.] <http://www.drop-ingenieria.es/scada.html>.
2. Patiño, Yuri VictorMunayev. *XStormClient, visualizador web para el SCADA Guardián del Alba*. 2011.
3. Penin, Aquilino Rodríguez. *Sistemas SCADA 2da Edicion*. Barcelona : MARACOBO S.A., 2007.
4. Lodoño, Johnatan Montoya. Monitoreo y visualización de temperaturas en el área de generación de frío de la planta de derivados lácteos Colanta San Pedro de los Milagros. *Biblioteca Digital Repositorio Institucional*. [En línea] 2009. [Citado el: 5 de 01 de 2014.] [http://www.bdigital.unal.edu.co/917/..](http://www.bdigital.unal.edu.co/917/)
5. ISA. *ISA-TR77.60.04 — Fossil Fuel Power Plant Human-Machine Interface — CRT Displays*. 1995. 1-55617-609-0.
6. —. *ANSI/ISA-18.2-2009 Management of Alarm Systems*. 2009. 978-1-936007-19-6.
7. UNESCO. *Directrices de la UNESCO para las políticas de aprendizaje móvil*. Francia : UNESCO, 2013.
8. Márquez, Pedro Luis Martín, Haba, José Ramón Oliva y Zarco, Custodia Manjavacas. *MONTAJE Y MANTENIMIENTO DE EQUIPOS*. Madrid : Ediciones Parainfo, S.A., 2010.
9. Caballero, Joaquín Molina y Espejo, Laura Baena. *Sistemas Operativos en Entornos monousuario y multiusuario*. Madrid : Vision Net, 2007.
10. Google. www.android.com. [En línea] Google. [Citado el: 15 de enero de 2014.] <http://developer.android.com/about/index.html>.
11. Apple. www.apple.com. [En línea] Apple. [Citado el: 12 de diciembre de 2013.] <https://www.apple.com/es/ios/what-is/>.
12. Blackberry. es.blackberry.com. [En línea] Blackberry. [Citado el: 15 de enero de 2014.] <http://es.blackberry.com/software/smartphones/blackberry-10-os.html>.
13. mySCADA Technologies. www.myscada.org. [En línea] mySCADA Technologies. [Citado el: 25 de diciembre de 2013.] <http://www.myscada.org/wp/myscada-2/>.
14. ProSoft Technology, Inc. itunes.apple.com. [En línea] ProSoft Technology, Inc, 14 de enero de 2014. [Citado el: 5 de marzo de 2014.] <https://itunes.apple.com/us/app/prosoft-i-view/id385920646?mt=8>.



15. SweetWilliam. www.sweetwilliams.com. [En línea] SweetWilliam. [Citado el: 25 de diciembre de 2013.] <http://www.sweetwilliams.com/scadamobile/features/>.
16. Greco, Peter Lubbers & Frank. WebSocket.org. [En línea] [Citado el: 25 de noviembre de 2013.] WebSocket.org.
17. Cámbar, Julio César Moreno. *XStormServer, servidor para el cliente de visualización web del SCADA Guardián del ALBA*. Habana : s.n., 2012.
18. Digia. qt.gitorious.org. [En línea] Qt Digia. [Citado el: 19 de diciembre de 2013.] <https://qt.gitorious.org/qt/qtwebsockets/>.
19. Tavendo GmbH. Autobahn | Testsuite. [En línea] [Citado el: 5 de diciembre de 2013.] <http://autobahn.ws/testsuite/>.
20. Digia. qt-project.org. [En línea] Qt Digia. [Citado el: 20 de noviembre de 2013.] <http://qt-project.org/doc/qt-4.8/qtquick.html>.
21. —. qt-project.org. [En línea] Qt Digia. [Citado el: 20 de noviembre de 2013.] <https://qt-project.org/wiki/Category:Tools::QtCreator>.
22. —. qt-project.org. [En línea] Qt Digia. [Citado el: 21 de noviembre de 2013.] <http://qt-project.org/doc/qt-5.0/qtqml/qtqml-index.html>.
23. Halterman, Richard L. *Fundamentals of C++ Programming*. 2014.
24. Elliott, Eric. *Programming JavaScript Applications*. s.l. : O'Reilly, 2013 .
25. Visual Paradigm International Ltd. www.visual-paradigm.com. [En línea] Visual Paradigm International Ltd. [Citado el: 18 de enero de 2014.] [paradim http://www.visual-paradigm.com/product/vpuml/](http://www.visual-paradigm.com/product/vpuml/).
26. López, Francisco J. Toro. *Administración de proyectos de informática*. Bogota : ECOE EDICIONES, 2012.
27. Beck, Kent. *Extreme Programming Explained*. 1999.
28. Pérez, Beatriz. *GXP – Adaptación y Aplicación de eXtreme Programming*. Montevideo, Uruguay : Universidad de la República, Grupo de Ingeniería de Software. 11000.
29. Letelier, Patricio y Penadés, Maria Carmen. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. s.l. : Universidad Politécnica de Valencia.
30. Beck, Kent. *Extreme Programming Explained: Embrace Change*. s.l. : Addison-Wesley Professional, 2004. ISBN-10: 0321278658.



31. JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. *El Proceso Unificado de Desarrollo de Software*. s.l. : Pearson Educación, 2000. ISBN: 84-7829-036-2.
32. Beck, Kent. *Planning Extreme Programming*. s.l. : Addison-Wesley Professional, 2000. ISBN-10: 0201710919.
33. IEEE. *[IEEE 1471]Standard Glossary of Software Engineering Terminology*.
34. Avendaño, Bertha Mariel Márquez y Rugarcía, José Manuel Zulaica. *Implementación de un reconocedor de voz gratuito a el sistema de ayuda a invidentes Dos-Vox en español*. Puebla : s.n., 2004.
35. Digia. qt-project. [En línea] Qt Digia. [Citado el: 15 de abril de 2014.] <http://qt-project.org/doc/qt-4.8/model-view-programming.html>.
36. Larman, Craig. *UML y Patrones 2da Edición*. s.l. : PEARSON EDUCACION, 2003. 8420534382.
37. Berzal, Fernando. *Apuntes de programación orientada a objetos en Java: Fundamentos de programación y principios de diseño*. 2011. ISBN 84-611-1405-1.

