



**Título: “Cliente HMI del SCADA Guardián del Alba para
Dispositivos Móviles.”**

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

Autor(es): José Enrique Díaz

Ernesto Martínez Riverón

Tutor(es): Ing. Leonardo Rafael Fernández Ruiz

Ing. Yosvani Ramírez Martínez

La Habana, junio del 2014.

“Año del 56 Aniversario del Triunfo de la Revolución”

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

José Enrique Díaz Ramos

Ernesto Martínez Riverón

Firma del Autor

Firma del Autor

Ing. Leonardo Rafael Fernández

Ing. Yosvani Ramírez Martínez

Firma del Tutor

Firma del Tutor

Datos de contacto

Ing. Leonardo Rafael Fernández Ruiz.

Ingeniero Informático se ha desempeñado como profesor, desarrollador e investigador, en la Línea de Sistemas Embebidos. Ha estado vinculado al área productiva al desarrollo de sistema de Realidad Virtual, Sistemas Empotrados, Sistemas de Gestión Industrial y SCADAs con 5 años de experiencia.

e-mail: lfernandez@uci.cu

Ing. Yosvani Ramírez Martínez.

Graduado de la especialidad de Ingeniero en Ciencias Informáticas desde hace un año, recién graduado en adiestramiento.

e-mail: yramz@uci.cu

Agradecimientos

José Enrique Díaz Ramos

A mis padres por su inmenso apoyo e incansable guía a lo largo de toda mi vida.

A mi familia por estar ahí siempre que los necesité.

A mi novia quien me ayudó muchísimo en la tesis.

A mis amigos y compañeros de aula, sin los cuales hoy no estuviera aquí.

A los profesores que me transmitieron su conocimiento a lo largo de estos años como estudiante.

A mi tutor que con sus ideas ayudó a mejorar este trabajo.

A todos aquellos que de una forma u otra me ayudaron a estar aquí hoy, ya sea con una palabra de aliento o un consejo, a todos, gracias.

Ernesto Martínez Riverón

A toda mi familia en especial a mi mamá por su apoyo durante todo este tiempo.

A la Revolución y a la UCI por darme la oportunidad única de convertirme en profesional.

A todos aquellos amigos, que han recorrido el mismo camino durante estos años.

A los profesores por brindarnos sus conocimientos y a todo aquel que contribuyó a esta investigación, en especial a mis amigos del 88204 por prestarnos sus teléfonos.

Resumen

La automatización de los procesos en la industria aparejado al desarrollo de las Tecnologías de la Informática y las Comunicaciones¹ han permitido la utilización de sistemas de adquisición de datos para controlar los procesos llevados a cabo en las industrias, estos sistemas se conocen como SCADA. En la actualidad se presenta una tendencia en el desarrollo de estos sistemas a utilizar la tecnología móvil como medio complementario para visualizar la información supervisada, debido al desarrollo que ha alcanzado la misma en los últimos años.

Este trabajo surge de la necesidad de tener acceso constante a la información generada por el Sistema SCADA 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 que permita visualizar en dispositivos móviles la información supervisada por el Sistema SCADA Guardián del Alba. Para el cumplimiento del mismo se realiza una descripción de algunas de las aplicaciones similares en el ámbito internacional, se refleja una investigación acerca de los principales aspectos y estándares a tener en cuenta a la hora de diseñar clientes HMI, así como una caracterización de la tecnología móvil en la actualidad. 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, tecnología móvil, HMI, accesibilidad.

¹ Tecnologías de la Informática y las Comunicaciones: Más conocidas como TIC por sus siglas en español.

Índice

| | |
|---|-----------|
| Introducción..... | 11 |
| 1 Capítulo 1: Fundamentación teórica..... | 15 |
| 1.1 Introducción..... | 15 |
| 1.2 Sistemas SCADA..... | 15 |
| 1.3 HMI..... | 16 |
| 1.4 Tecnología móvil..... | 20 |
| 1.4.1 Teléfono inteligente..... | 20 |
| 1.4.2 Sistemas operativos para móviles..... | 21 |
| 1.4.3 Características que afectan el diseño de interfaces en dispositivos móviles:..... | 23 |
| 1.4.4 Aplicaciones similares..... | 24 |
| 1.5 Selección de tecnologías de desarrollo y comunicación..... | 26 |
| 1.5.1 Protocolo WebSocket..... | 26 |
| 1.5.2 QTWebsocket..... | 27 |
| 1.5.3 Plataformas de desarrollo para móviles..... | 27 |
| 1.6 Ambiente de Desarrollo Integrado (IDE)..... | 28 |
| 1.6.1 QT Creator..... | 28 |
| 1.7 Lenguajes de programación..... | 29 |
| 1.7.1 QML..... | 29 |
| 1.7.2 C++..... | 30 |
| 1.7.3 JavaScript..... | 30 |
| 1.8 Herramientas CASE..... | 31 |
| 1.8.1 Visual Paradigm..... | 31 |
| 1.9 Metodología de Desarrollo de Software..... | 32 |
| 1.9.1 Programación extrema (XP)..... | 32 |
| 1.10 Consideraciones parciales..... | 34 |
| 2 Capítulo 2: Solución propuesta..... | 36 |
| 2.1 Introducción..... | 36 |
| 2.2 Propuesta del sistema..... | 36 |

| | | |
|------------|--|----|
| 2.3 | Fase de exploración y planificación | 37 |
| 2.3.1 | Historias de usuario | 37 |
| 2.3.2 | Requisitos no funcionales | 45 |
| 2.3.3 | Estimación de esfuerzo por historias de usuario..... | 46 |
| 2.3.4 | Plan de iteraciones | 47 |
| 2.3.5 | Plan de entregas..... | 47 |
| 2.4 | Diseño de la solución propuesta | 48 |
| 2.4.1 | Arquitectura | 48 |
| 2.4.2 | Patrones de diseño | 51 |
| 2.4.3 | Diagrama de clases..... | 53 |
| 2.4.4 | Tarjetas CRC..... | 54 |
| 2.4.5 | Diagrama de despliegue..... | 57 |
| 3 | Capítulo 3: Validación de la propuesta | 58 |
| 3.1 | Introducción..... | 58 |
| 3.2 | Fase de pruebas. | 58 |
| 3.2.1 | Pruebas de aceptación | 58 |
| 3.3 | Consideraciones parciales | 71 |
| | Conclusiones | 72 |
| | Recomendaciones | 73 |
| | Bibliografía | 74 |
| | Glosario de términos | 77 |

Índice de figuras

| | |
|---|-----------|
| <i>Figura 1: Teléfono Inteligente</i> | <i>21</i> |
| <i>Figura 2: mySCADA.....</i> | <i>24</i> |
| <i>Figura 3: Prosoft i-View.....</i> | <i>25</i> |
| <i>Figura 4: ScadaMobile</i> | <i>25</i> |
| <i>Figura 5: Propuesta de Sistema.....</i> | <i>36</i> |
| <i>Figura 6: Patrón Cliente-Servidor.....</i> | <i>49</i> |
| <i>Figura 7: Patrón Cliente-Servidor en la aplicación.</i> | <i>49</i> |
| <i>Figura 8: Patrón Modelo-Vista</i> | <i>50</i> |
| <i>Figura 9: Patrón Modelo-Vista en la aplicación.</i> | <i>51</i> |
| <i>Figura 10: Patrón Controlador en la aplicación.</i> | <i>52</i> |
| <i>Figura 11: Patrón Creador en la aplicación.</i> | <i>53</i> |
| <i>Figura 12: Diagrama de clases.</i> | <i>54</i> |
| <i>Figura 13: Diagrama de despliegue.</i> | <i>57</i> |
| <i>Figura 14: Resultados de las pruebas de aceptación.....</i> | <i>70</i> |

Índice de tablas

| | |
|---|----|
| Tabla 1: Tratamiento de colores en el SCADA Guardián del alba..... | 17 |
| Tabla 2: Historia de usuario: Autenticar Usuario..... | 37 |
| Tabla 3: Tarea: Conexión con el servidor..... | 38 |
| Tabla 4: Autenticarse en el servidor..... | 38 |
| Tabla 5: Historia de usuario: Mostrar sumario de alarmas..... | 38 |
| Tabla 6: Tarea: Implementar interfaz del sumario de Alarmas..... | 38 |
| Tabla 7: Tarea: Filtrar alarmas..... | 39 |
| Tabla 8: Historia de Usuario: Mostrar sumario de señales analógicas..... | 39 |
| Tabla 9: Tarea: Implementar interfaz del sumario de señales analógicas..... | 39 |
| Tabla 10: Tarea: Filtrar señales analógicas..... | 40 |
| Tabla 11: Tarea: Gráficas de tendencia para señales analógicas..... | 40 |
| Tabla 12: Historia de Usuario: Mostrar sumario de señales digitales..... | 40 |
| Tabla 13: Tarea: Implementar interfaz del sumario de señales digitales..... | 41 |
| Tabla 14: Tarea: Filtrar señales digitales..... | 41 |
| Tabla 15: Tarea: Gráficas de tendencia para señales digitales..... | 41 |
| Tabla 16: Historia de Usuario: Mostrar eventos..... | 41 |
| Tabla 17: Implementar interfaz de sumario de Eventos..... | 42 |
| Tabla 18: Tarea: Filtrar eventos..... | 42 |
| Tabla 19: Historia de Usuario: Monitor del sistema..... | 42 |
| Tabla 20: Tarea: Mostrar estado de los módulos..... | 43 |
| Tabla 21: Tarea: Implementar interfaz de sumario de subcanales..... | 43 |
| Tabla 22: Tarea: Filtrar subcanales..... | 43 |
| Tabla 23: Tarea: Implementar interfaz de sumario de dispositivos..... | 44 |
| Tabla 24 : Tarea: Filtrar dispositivos..... | 44 |
| Tabla 25: Historia de Usuario: Proveer ayuda..... | 44 |
| Tabla 26: Actores..... | 45 |
| Tabla 27: Estimación de esfuerzo por historias de usuario..... | 46 |
| Tabla 28: Plan de iteraciones..... | 47 |
| Tabla 29: Plan de entregas..... | 48 |
| Tabla 30: Tarjeta CRC para la clase Conexion..... | 55 |

| | |
|--|----|
| Tabla 31: Tarjeta CRC para la clase Alarms. | 55 |
| Tabla 32: Tarjeta CRC para la clase Points. | 55 |
| Tabla 33: Tarjeta CRC para la clase User..... | 56 |
| Tabla 34: Tarjeta CRC para la clase Device. | 56 |
| Tabla 35: Tarjeta CRC para la clase Events. | 56 |
| Tabla 36: Tarjeta CRC para la clase Monitor. | 56 |
| Tabla 37: Tarjeta de aceptación "Conectar con el servidor". | 58 |
| Tabla 38: Tarjeta de aceptación "Autenticar usuario". | 59 |
| Tabla 39: Tarjeta de aceptación "Cerrar sesión". | 59 |
| Tabla 40: Tarjeta de aceptación "Mostrar sumario de alarmas". | 60 |
| Tabla 41: Tarjeta de aceptación "Filtrar alarmas". | 60 |
| Tabla 42: Tarjeta de aceptación "Mostrar detalles de alarma". | 61 |
| Tabla 43: Tarjeta de aceptación "Mostrar señales analógicas". | 61 |
| Tabla 44: Tarjeta de aceptación "Filtrar señales analógicas". | 62 |
| Tabla 45: Tarjeta de aceptación "Mostrar detalles de señales analógicas". | 62 |
| Tabla 46: Tarjeta de aceptación "Mostrar señales digitales". | 63 |
| Tabla 47: Tarjeta de aceptación "Filtrar señales digitales". | 63 |
| Tabla 48: Tarjeta de aceptación "Mostrar detalles de señales digitales" | 64 |
| Tabla 49: Tarjeta de aceptación "Mostrar sumario de eventos". | 64 |
| Tabla 50: Tarjeta de aceptación "Filtrar eventos". | 65 |
| Tabla 51: Tarjeta de aceptación "Mostrar detalles de eventos". | 66 |
| Tabla 52: Tarjeta de aceptación "Mostrar sumario de subcanales". | 66 |
| Tabla 53: Tarjeta de aceptación "Filtrar subcanales". | 67 |
| Tabla 54: Tarjeta de aceptación "Mostrar detalles de subcanal". | 67 |
| Tabla 55: Tarjeta de aceptación "Mostrar sumario de dispositivos". | 68 |
| Tabla 56: Tarjeta de aceptación "Filtrar dispositivos". | 68 |
| Tabla 57: Tarjeta de aceptación "Mostrar detalles de dispositivos". | 69 |
| Tabla 58: Tarjeta de aceptación "Proveer ayuda". | 69 |

Introducción

La automatización de los procesos en la industria aparejado al desarrollo de las Tecnologías de la Informática y las Comunicaciones (TIC) han permitido la utilización de sistemas de adquisición de datos para controlar los procesos llevados a cabo en las industrias, estos sistemas se conocen como SCADA siglas en inglés de *Supervisory Control and Data Acquisition*. Un SCADA es un software que permite controlar y supervisar procesos industriales a distancia, facilitando la retroalimentación en tiempo real con los dispositivos de campo (sensores y actuadores). Además provee toda la información que se genera en la producción permitiendo su gestión e intervención (1). Estos sistemas están generalmente compuestos por los siguientes módulos: Base de Datos Históricas, Middleware, Seguridad, Reportes, Interfaz Hombre Máquina², Núcleo de procesamiento de datos, Manejadores y Configuración.

La visualización de los procesos tecnológicos e industriales es clave para los sistemas SCADA, ya que los operadores necesitan visualizar representaciones del proceso de producción de forma realista, con buena presencia y simple de operar; dando la posibilidad de visualizar la información general del proceso, para garantizar una respuesta rápida del operador ante eventos generados por el sistema. En la Universidad de las Ciencias Informáticas existe el Centro de Informática Industrial (CEDIN) que se dedica al desarrollo de un Sistema SCADA llamado Sistema Integral de Supervisión y Control de Procesos Guardián del Alba. El mismo tiene como propósito automatizar los procesos industriales de pequeñas, medias y grandes industrias que requieran del control de sus procesos, actualmente se encuentra en fase de implantación en varias instalaciones de PDVSA en la República Bolivariana de Venezuela.

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

² Interfaz Hombre Máquina: Conocido como HMI de las siglas en inglés de Human Machine Interface.

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.

Hoy en día, los celulares actúan como otro apéndice para muchas personas, se usan para llamar, enviar correos electrónicos o navegar por Internet. Los nuevos dispositivos móviles permiten acceder a multitud de servicios, convirtiéndose en pequeños ordenadores a través de los cuales se interactúa con un entorno cada vez más tecnológico. Conceptos como web móvil o geo-localización, están directamente relacionados con este tipo de tecnologías a las que no se puede dar la espalda, puesto que son las que una parte creciente de los usuarios utilizan de manera habitual.

Las características de este tipo de dispositivos 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 la accesibilidad de la información, 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 (2) 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. 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 **problema de la investigación**: El Sistema SCADA Guardián del Alba no cuenta con un medio eficiente para la visualización en dispositivos móviles, limitando la toma de decisiones del proceso productivo.

Por lo que el **objeto de estudio** es: El proceso de visualización realizado por el Sistema SCADA Guardián del Alba, teniendo como **campo de acción** la visualización de la información del SCADA Guardián del Alba para dispositivos móviles.

Definiendo como **objetivo general** desarrollar un cliente HMI que permita visualizar en dispositivos móviles, la información supervisada por el Sistema SCADA Guardián del Alba.

La **idea a defender** con el desarrollo de un cliente HMI para dispositivos móviles en el proceso de visualización del sistema SCADA Guardián del Alba se contribuirá a una mayor accesibilidad a la información supervisada por el Sistema SCADA Guardián del Alba para la toma de decisiones del proceso productivo.

El desarrollo de esta investigación estará guiado por las siguientes tareas:

- 1- Conceptualización de los elementos que sostienen el desarrollo de un cliente HMI para la visualización de mediciones industriales.
- 2- Implementación de un cliente HMI para dispositivos móviles para el Sistema SCADA Guardián del Alba.
- 3- Validación de la propuesta de solución de acuerdo a los requerimientos propuestos.

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

Métodos Teóricos:

Método histórico-lógico: Para la comprensión de los antecedentes y las tendencias actuales de los sistemas SCADA y las tecnologías de desarrollo para dispositivos móviles.

Método analítico-sintético: Para el estudio de los conceptos empleados en los sistemas SCADA, analizando los documentos elaborados por desarrolladores, para la extracción de los elementos más importantes.

Con esta investigación se pretende implementar un cliente HMI para dispositivos móviles que permita visualizar sumarios de alarmas, eventos, señales (digitales y analógicas), dispositivos y subcanales, gráficas de tendencia y que permita conocer el estado de los módulos, satisfaciendo la necesidad de información en un proceso que esté siendo supervisado por un sistema SCADA de los usuarios que necesiten la información en tiempo real y no tienen acceso a las salas de control.

Este trabajo de diploma está organizado de la siguiente forma:

Capítulo 1 Fundamentación Teórica: En este capítulo se hace referencia a los sistemas SCADA, así como a los estándares y buenas prácticas a la hora de diseñar un HMI y el estado del arte de las tecnologías actuales de desarrollo para dispositivos móviles y las tecnologías de comunicación.

Capítulo 2 Solución propuesta: En este capítulo se definen las funcionalidades y se construye la propuesta de solución, basado en las historias de usuario como artefacto conductor de la metodología XP.

Capítulo 3 Validación de la propuesta: En este capítulo se realizan las pruebas al software con el objetivo de satisfacer los requerimientos del sistema con calidad y de acuerdo al objetivo general de la investigación.

1 Capítulo 1: Fundamentación teórica

1.1 Introducción

En este capítulo se detallan las principales características de los sistemas SCADA profundizando en el módulo HMI, del cual se explican sus funciones fundamentales y las aplicaciones que lo componen. Unido a esto se realiza un profundo estudio de las tecnologías para desarrollar aplicaciones orientadas a dispositivos móviles.

1.2 Sistemas SCADA

Se nombra SCADA a cualquier software que permita el acceso a datos remotos de un proceso y posibilite, utilizando las herramientas de comunicación necesarias en cada caso, el control del mismo.

No se trata de un sistema de control, sino de un software de monitorización o supervisión, que realiza la tarea de interfaz entre los niveles de control (PLC) y los de gestión, a un nivel superior (3).

Requisitos comunes de los Sistemas SCADA.

- Arquitectura abierta que permita combinaciones con aplicaciones estándar y de usuario, que brinden la posibilidad a los integradores de crear soluciones de mando y supervisión optimizadas.
- Sencillez de instalación, sin exigencias de hardware elevadas, fácil de utilizar y con interfaces amigables con el usuario.
- Permite la integración con las herramientas ofimáticas y de producción.
- Fácilmente configurable y escalable, debe ser capaz de crecer o adaptarse según las necesidades cambiantes de la empresa.
- Funciones de mando y supervisión integradas.

Ventajas de los Sistemas SCADA

- Economía: es más fácil saber lo que ocurre en la instalación desde la oficina que enviar a un operario a realizar la tarea. Ciertas revisiones se convertirán en innecesarias.
- Accesibilidad: Será posible modificar los parámetros de funcionamiento de la fábrica a distancia.

- **Mantenimiento:** la adquisición de datos materializa la posibilidad de obtener datos de un proceso. Esta puede ser programada de manera que informe cuando se acerquen las fechas de revisión o cuando una máquina tenga más fallos de los considerados normalmente.
- **Gestión:** todos los datos recopilados pueden ser valorados de múltiples maneras mediante herramientas estadísticas, gráficas, valores tabulados, entre otros que posibiliten un mejor rendimiento del sistema (3).

1.3 HMI

Esta interfaz, muestra los datos de uno o varios procesos a un operador (humano) y además permite al operario ejercer el control sobre los mismos. Los HMI surgen esencialmente de la necesidad de estandarizar la manera de monitorear y de controlar múltiples sistemas remotos como los PLC (Programmable Logic Controller) y otros mecanismos de control. Aunque un PLC realiza automáticamente un control pre-programado sobre un proceso, normalmente se distribuye 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 un modo 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 algún tipo de red, a continuación 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 (4).

Aplicaciones que componen un HMI

- **Ambiente de configuración**

Permite configurar varios procesos o parte de ellos, posibilitando al usuario definir el entorno de trabajo de su aplicación, según la disposición de pantallas requeridas 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 controladores de comunicación que permitirán el enlace con los elementos de campo y la posible conexión 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. (3)

Directrices de codificación de colores:

- Mientras que 11 colores pueden ser distinguidos en un diseño óptimo, se deben utilizar no más de cuatro o cinco colores, ya que a medida que el número de colores aumenta, la eficacia relativa de los códigos de colores disminuye.
- Usar colores brillantes (por ejemplo, verde o amarillo claro) para resaltar los elementos tales como datos dinámicos o símbolos dinámicos; utilizar colores oscuros (por ejemplo, rojo o índigo) para reprimir elementos estáticos.
- Evitar azul o rojo oscuro en símbolos o caracteres alfanuméricos. Estos colores muestran bajo brillo cuando se utilizan contra un fondo de pantalla oscuro.
- Los fondos de pantalla deben ser oscuros. Los elementos de la pantalla con colores brillantes tendrán mejor contraste con un fondo oscuro (El 25 a 50 por ciento gris es el mejor, pero elementos específicos pueden utilizar fondos más oscuros para mejorar el contraste) (5).

Uno de los aspectos más importantes en los HMI es el tratamiento de alarmas, la siguiente tabla muestra el tratamiento de alarmas utilizado en el SCADA Guardián del ALBA, el mismo está basado en la norma ISA 18.2 (6).

Tabla 1: Tratamiento de colores en el SCADA Guardián del alba.

| | Representación en Sumario según valor de severidad | | |
|--------------------------------|--|---|---|
| Estado de Alarma | Severidad 1 o Crítica | Severidad 2 o Media | Severidad 3 Baja |
| Alarma Activa No Reconocida | Fondo Negro, Letras Rojas, intermitencia hacia | Fondo Negro, Letras Naranja, intermitencia hacia | Fondo Negro, Letras Amarillas, intermitencia hacia |

| | | | |
|--------------------------------|---|--|---|
| | Fondo Rojo, Letras Negras cada 0,5 Seg. | Fondo Naranja, Letras Negras cada 0,5 Seg | Fondo Amarillo, Letras Negras cada 0,5 Seg. |
| Alarma Activa Reconocida | Fondo Negro, Letras Rojas, en modo estático. | Fondo Negro, Letras Naranja, en modo estático. | Fondo Negro, Letras Amarillas, en modo estático |
| Alarma No Activa No reconocida | Fondo Negro, Letras Rojas, intermitencia hacia Fondo Negro, Letras Verdes cada 0,5 Seg. | Fondo Negro, Letras Naranjas, intermitencia hacia Fondo Negro, Letras Verdes cada 0,5 Seg. | Fondo Negro, Letras Rojas, intermitencia hacia Fondo Negro, Letras Verdes cada 0,5 Seg. |
| Alarmas No Activas reconocidas | Desaparecen del sumario | | |

Técnicas de entrada del operador:

Los dispositivos de entrada permiten al operador introducir comandos u otra información al sistema de control. Los dispositivos de entrada más utilizados en la actualidad son la pantalla táctil, el mouse (o trackball) y el teclado. El uso de dispositivos de entrada en pantalla (mouse o pantalla táctil) mantiene la atención del operador sobre los datos, y por lo tanto minimizar los errores de manejo. Este método es preferido sobre el teclado. La respuesta del sistema a las entradas del operador debe ser positiva, obvia y natural, el sistema siempre debe responder a las aportaciones recibidas. El tiempo de respuesta debe ser instantáneo (menos de 2 segundos). Si el tiempo de respuesta es superior a 2 segundos, la retroalimentación instantánea debe mostrar que el proceso está en marcha (5).

- Pantalla táctil:

Una de las ventajas de una pantalla táctil, en comparación con el uso de un ratón, es el movimiento automático del cursor a la posición deseada. Las desventajas son que los objetivos significativamente más grandes están obligados a ser "tocados"

constantemente y que las manchas de huellas digitales pueden afectar la legibilidad de la pantalla.

Directrices de diseño para pantallas táctiles:

1. Proporcionar áreas objetivo bien marcadas.
2. Proporcionar la diferenciación entre los objetivos para que sean fácilmente distinguible la función que realizan.
3. Resaltar la selección del usuario cuando se realiza la misma.
4. Utilizar múltiples acciones táctiles para el control (recomendado) para evitar el contacto accidental (5).

Reglas generales de organización del contenido:

1. Proporcionar el contenido necesario, para lograr el objetivo de la pantalla.
2. Organizar el contenido de manera que no sea demasiado denso. Ya que pantallas cargadas hacen difícil localizar la información. Una regla de oro es proporcionar un mínimo de 25 a 40 por ciento de espacio en blanco. Un diseño de pantalla eficaz permitirá que más información se pueda incluir en una sola pantalla sin estorbar.
3. Organizar los elementos en el orden en que serán utilizados (por ejemplo, de arriba a abajo, de izquierda a derecha), los elementos más frecuentemente utilizados se muestran con más protagonismo (5).

Consejos de diseño para evitar proveer informaciones erróneas:

- Utilizar los mismos colores estandarizados que los del proceso que se está monitorizando, de manera tal que las diferencias de interpretación sean pequeñas y el usuario tenga que aprender el mínimo número de novedades.
- Utilizar combinaciones de elementos como el color y texto o color y forma.
- No almacenar indicadores con colores primarios, ya que esto provoca la aparición de colores complementarios en la imagen retenida en la retina, lo cual puede hacer que se vean colores fantasmas.
- Clasificar todo lo que sucede en el sistema en dos categorías alarmas y eventos.
- Debido a la existencia de trastornos de visión asociados a los colores, cualquier cambio de color debería estar complementado por otros indicadores inequívocos (cambio de forma, textos o emisión de sonidos) (3).

Principales retos a cumplir en el desarrollo de Interfaces Hombre-Máquina:

- Disminuir los errores de los operarios utilizando representaciones claras e intuitivas de las interfaces de control.
- Reducir los tiempos de aprendizaje de los nuevos operadores y conseguir que los tiempos de formación sean acumulativos, permitiendo el cambio de un sistema a otro con el mínimo entrenamiento, gracias a la estandarización de las interfaces de control.
- Reducir costes de rediseño (3).

1.4 Tecnología móvil

Las tecnologías móviles están en constante evolución, la diversidad de dispositivos existentes es inmensa e incluye, a grandes rasgos, los teléfonos móviles, las tabletas, los lectores electrónicos, los reproductores de sonido portátiles y las consolas de juegos manuales. Por lo tanto las tecnologías móviles son aquellas que son digitales, portátiles, controladas por lo general por una sola persona, tienen acceso a internet y capacidad multimedia, y pueden facilitar un gran número de tareas, especialmente las relacionadas con la comunicación (7).

Con el crecimiento actual de este sector tecnológico se han experimentado diversos avances, apareciendo así el concepto de teléfono inteligente, el cual permite realizar una inmensa gama de acciones, además de las habituales.

1.4.1 Teléfono inteligente

Un teléfono inteligente es un dispositivo electrónico que hace las funciones de un teléfono móvil, a la vez que incorpora características similares a las de un ordenador de mano.

Todos ellos soportan un cliente de correo con la funcionalidad completa de un organizador personal, así como el acceso a internet. Una de las características importantes de todos los teléfonos inteligentes es que permiten la instalación de programas para aumentar sus posibilidades. Se le pueden instalar juegos, programas de comunicación, etc (8). Estos dispositivos tienen generalmente el aspecto de la Figura 1.



Figura 1: Teléfono Inteligente. Fuente: www.apple.com

Características

- Utiliza un sistema operativo.
- Tiene GPS.
- Dispone de varias opciones para insertar datos, como la pantalla táctil o el teclado QWERTY.
- Permite conectarse a Internet.
- Permite instalar aplicaciones desde terceros.
- Permite visualizar documentos en casi cualquier formato, como los PDFs u hojas de cálculo.

1.4.2 Sistemas operativos para móviles

Un sistema operativo es un programa destinado a permitir la comunicación del usuario con el computador y gestionar sus recursos de una forma eficaz. Comienza a trabajar cuando se enciende el computador, y gestiona el hardware del dispositivo desde los niveles más básicos, así como el software instalado en este, permitiendo la utilización del software de manera cómoda (9).

Inicialmente este tipo de software fue desarrollado para las computadoras pero en la actualidad son utilizados en los teléfonos celulares para tener esa misma interlocución entre el hardware y los programas que el usuario desea utilizar. Entre los ejemplos de sistemas operativos para celulares se encuentran: Android, iOS, Blackberry OS y otros desarrollados específicamente para una u otra marca de celulares.

Android:

Android es un sistema operativo para dispositivos móviles y es el más difundido a nivel mundial. Está basado en GNU/Linux e inicialmente fue desarrollado por Google. La presentación de la plataforma Android se realizó el 5 de noviembre de 2007 junto con la fundación Open Handset Alliance, un consorcio de 48 compañías de hardware, software y telecomunicaciones comprometidas a la promoción de estándares abiertos para dispositivos móviles.

Android proporciona una plataforma de nivel mundial que permite el desarrollo de aplicaciones y juegos para Android. Para ayudar en el desarrollo eficiente provee el Android Developer Tools (ADT), el cual contiene un Ambiente de Desarrollo Integrado (IDE) para el lenguaje de programación Java con funciones avanzadas para desarrollar, depurar y empaquetar aplicaciones para Android. Usando el IDE, se puede desarrollar para cualquier dispositivo Android disponible o crear dispositivos virtuales que emulan cualquier configuración de hardware. La mayoría del código fuente de Android ha sido publicado bajo la licencia de software Apache, una licencia de software libre y código fuente abierto (10).

iOS:

iOS es el sistema operativo que da vida a dispositivos como el iPhone, el iPad, el iPod Touch o el Apple TV. La simplicidad y optimización son sus pilares haciendo que millones de usuarios se decanten por iOS en lugar de escoger otras plataformas que necesitan más hardware para mover con fluidez el sistema. Cada año, Apple lanza una gran actualización de iOS que suele traer características exclusivas para los dispositivos más sobresalientes que estén a la venta en ese momento.

Características generales del iOS

- Su interfaz gráfica está diseñada para pantallas táctiles, con capacidad para gestos multitouch.
- Su interfaz está constituida básicamente por deslizadores, interruptores y botones, con una respuesta inmediata y fluida.
- Soporta acelerómetros internos al dispositivo.
- Emplea unos 500 MB de almacenamiento, aunque varía de modelo en modelo.
- Incluye múltiples aplicaciones para gestionar correos electrónicos, fotos, cámara, mensajes, clima, notas, YouTube, contactos, reloj, entre otras.

- Soporta multitarea (con algunas limitaciones) (11).

Blackberry OS:

El BlackBerry OS es un sistema operativo para móviles desarrollado por Research In Motion (RIM) para su línea de Smartphone (teléfono inteligente) BlackBerry. La plataforma BlackBerry es muy conocida por su soporte nativo al correo electrónico corporativo a través de MIDP, que permite activación inalámbrica completa y sincronización con Microsoft Exchange, Lotus Domin, o Novell GroupWise. Los desarrolladores ajenos a la empresa pueden escribir software empleando las clases BlackBerry API.

Características generales de BlackBerry OS

- Soporta multitarea.
- Soporta dispositivos especiales como trackballs, trackpad, touchscreen y trackwheel (12).

1.4.3 Características que afectan el diseño de interfaces en dispositivos móviles:

- **Uso en exteriores**

La luz solar directa sobre la pantalla afecta a la legibilidad por lo tanto el contraste entre los elementos del diseño es clave.

- **Pantallas pequeñas**

Las pantallas siempre serán más pequeñas que la de un ordenador, por lo que se ven menos líneas de texto y menos caracteres por línea. Una imagen demasiado grande puede lanzar el contenido por debajo de la línea de scroll y en los dispositivos no táctiles hacer scroll es un poco más complejo.

- **No se realizan tareas largas o complejas**

Diseña los procesos o tareas de forma que no requieran mucho esfuerzo.

- **No están optimizados para la entrada de datos**

Los teclados son pequeños y muchas veces incompletos por lo que hay que dar opciones predefinidas siempre que sea posible.

- **Distinta percepción de privacidad y seguridad**

En aplicaciones webs financieras o que almacenan datos personales hay que saber transmitir al usuario que se encuentra en un área segura.

- **Se manejan con menor precisión**

En los dispositivos con pantallas táctiles se usan mucho los pulgares, pudiendo ser realmente difícil acertar en un elemento de interfaz determinado si es demasiado pequeño.

1.4.4 Aplicaciones similares

Algunos de los principales fabricantes de sistemas SCADA a nivel universal han portado sus aplicaciones hacia los dispositivos móviles. Ejemplo de esto son:

MySCADA

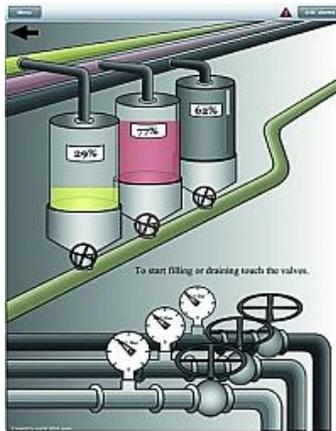


Figura 2: mySCADA. Fuente: mySCADA Technologies.

MySCADA es una aplicación SCADA con todas las funciones avanzadas de un operador. Es adecuado para los gerentes de planta y los operadores, supervisores, ingenieros de software, y personal de mantenimiento. Es compatible con iPads solamente. Puede ser utilizado para controlar, supervisar y mostrar los procesos tecnológicos, infraestructura, sistemas e instalaciones industriales. Muestra gráficas de tendencias básicas para monitorear las etiquetas, alarmas de monitoreo en línea para estados críticos y brinda opciones para la prioridad y el texto filtrado en la ventana de alarma (13).

ProSoft i-View



Figura 3: Prosoft i-View. Fuente: ProSoft Technology, Inc.

ProSoft i-View es desarrollado por la tecnología de ProSoft. Esta es una aplicación móvil HMI/SCADA que permite la monitorización remota y control de los valores de proceso dentro de una red EtherNet / IP y o Modbus TCP / IP, utilizando red inalámbrica o conexiones de red celular. La aplicación es ideal para el iPhone, iPod Touch y la versión para IPAD fue lanzado en septiembre de 2010. Al igual que otras aplicaciones SCADA se proporciona un panel de operador para el acceso, el seguimiento de las variables y la memoria de los PLCs. Mediante el uso de la plataforma de iOS, los ingenieros de la planta, los desarrolladores de software PLC y el personal de mantenimiento pueden monitorear y controlar sus sistemas en vivo desde cualquier lugar, las mediciones de flujo, las válvulas de control, los datos de proceso y los datos de estado (14).

ScadaMobile



Figura 4: ScadaMobile. Fuente: SweetWilliam, SL.

ScadaMobile es adecuado para el iPhone, iPod Touch y el IPAD. Es una aplicación simple HMI/SCADA para la supervisión de las variables y la memoria de Controladores Lógicos Programables (PLC) y Unidades de Terminal Remota (RTU). Esta aplicación es ideal para ingenieros de las plantas, los desarrolladores de software PLC, el personal de mantenimiento, y cualquier persona que trabaja con sistemas basados en PLC. Es compatible con las principales marcas de PLCs de EE.UU. y Japón y puede ser utilizado como un operador móvil local o remoto. Se relaciona con conexiones simultáneas a un número de PLCs y RTUs a distancia, e incluye funciones de automatización que facilitan la organización de un gran número de etiquetas. Para conectarse al PLC, ScadaMobile usa vínculos directos TCP / IP entre los dispositivos iOS y el PLC. Los usuarios introducen la dirección IP del PLC para establecer una conexión (15).

1.5 Selección de tecnologías de desarrollo y comunicación

A continuación se analizarán las soluciones técnicas que se utilizaron para la implementación del sistema realizando una descripción de las características y ventajas de estas soluciones.

1.5.1 Protocolo WebSocket

El protocolo WebSocket es una tecnología de comunicación que define los procedimientos para realizar una conexión mediante WebSocket usando TCP (Transmission Control Protocol). El mismo proporciona un canal de comunicación full-dúplex que opera a través de un único socket TCP.

Ventajas:

- Reduce en grandes proporciones el tráfico en la red, teniendo en cuenta que al establecer la comunicación WebSocket entre el cliente y el servidor solo hay un envío de 2 bits.
- Reduce la latencia ya que permite que se realicen conexiones en ambas direcciones al mismo tiempo.

Desventajas:

- Posible aparición de vulnerabilidades en el futuro ya que es un protocolo de reciente creación.
- Se necesita hacer trabajo adicional en el servidor. Esto se debe a que WebSocket hace su propia codificación binaria por encima de la TCP (16).

Esta tecnología se escogió teniendo en cuenta las características expuestas anteriormente, así como estudios realizados en investigaciones previas (17) donde se demostró que el protocolo WebSocket era una solución recomendable para realizar la comunicación, y además para garantizar la integración con el SCADA Guardián del ALBA.

1.5.2 QTWebSocket

QtWebSocket es una implementación del protocolo WebSocket realizada en C++, bajo la licencia GNU Lesser General Public License. La misma es muy útil, puesto que es compatible con navegadores como Google Chrome, Mozilla Firefox, Opera y Safari, además soporta la mayoría de los sistemas operativos actuales como son Windows, GNU Linux, MacOS, Android y IOS. Unido a esto posee una serie de funcionalidades entre las que se encuentran:

- Soporte para múltiples conexiones.
- Implementación del cliente y servidor.
- Protocolo WSS (18).

Esta biblioteca se escogió para garantizar la compatibilidad con el servidor actualmente en desarrollo el cual está siendo implementado con esta misma biblioteca y además la pruebas del sitio autobahn (19) demuestran la calidad de las conexiones del protocolo websocket mediante esta biblioteca.

1.5.3 Plataformas de desarrollo para móviles.

QtQuick

QtQuick es una aplicación multiplataforma y marco de interfaz de usuario para los desarrolladores que utilizan C++ o QML, CSS y JavaScript como lenguaje. Qt Creator es 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 crecientemente usadas en teléfonos móviles, tablets y otros dispositivos portables. QtQuick consta de un completo conjunto de elementos para interfaz de usuario y un lenguaje declarativo para describir interfaces de usuario. Además cuenta con una colección de APIs de C++ utilizadas para integrar funciones de alto nivel, creadas con su lenguaje declarativo con las aplicaciones clásicas de Qt. Sus herramientas de soporte se desarrollan como un proyecto de código abierto regido por un modelo meritocrático. Entre sus ventajas cuenta con componentes de interfaz de usuario de escritorio y táctiles, así como el ser

fácilmente estilizado, obteniendo una flexibilidad visual máxima. QtQuick se puede utilizar bajo código abierto (LGPL v2.1) o términos comerciales (20).

Se seleccionó ya que además de las características previamente expuestas da la posibilidad de realizar una aplicación multiplataforma y nativa al mismo tiempo, no encontrándose otra tecnología que ofrezca esta posibilidad, lo cual permitiría usar los recursos de hardware y software de los dispositivos. Además es compatible con el protocolo de comunicación escogido.

1.6 Ambiente de Desarrollo Integrado (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

1.6.1 QT Creator

Qt Creator es un IDE creado por Trolltech para el desarrollo de aplicaciones con las bibliotecas Qt. Está distribuido bajo los términos de GNU Lesser General Public License (y otras), es software libre y de código abierto.

Qt Creator incluye una amplia gama de herramientas útiles entre las que se encuentran:

- Smart Code Editor: es un editor de código que ofrece resaltado de sintaxis, así como de código.
- Qt Project Generating Wizard: este asistente permite al usuario generar un proyecto para una aplicación de consola, una aplicación con interfaz de usuario, o una biblioteca de C++.
- Qt Help Integration: se puede acceder fácilmente a toda la documentación haciendo clic en el botón de Ayuda (Help).
- Qt Designer Integration: la forma de la interfaz de usuario se puede diseñar fácilmente en Qt Creator. Simplemente se hace doble clic sobre un archivo “.ui” en el Explorador de proyectos para poner en marcha la integración.
- Locator: una potente herramienta de navegación que permite al usuario localizar los archivos y el uso mínimo de clases de teclado.

- Support for qmake's .pro file format: el archivo .Pro se usa como descripción del proyecto (21).

1.7 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

1.7.1 QML

El módulo de QtQML proporciona un framework para el desarrollo de aplicaciones y bibliotecas con el lenguaje QML. 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. Estos se brindan por el módulo QtQuick, que proporciona muchos componentes visuales, un framework para animación y mucho más para la construcción de interfaces de usuario con QML.

El lenguaje QML se usa principalmente para aplicaciones móviles, donde la entrada táctil, las animaciones fluidas y una buena experiencia de usuario son cruciales. Debido a esto y a las tendencias actuales, al uso de JavaScript y JSON y a la necesidad del uso de lenguajes declarativos que separen interfaz de usuario de codificación. QML es una versión adaptada para Qt de una tendencia hacia lenguajes declarativos en las interfaces de usuario. La gran diferencia en cuanto a tecnologías anteriores es la adopción de JSON en vez de XML.

Ventajas

- Velocidad de desarrollo.
- Impulsa la separación entre la presentación y la lógica.
- Una mejor integración entre los programadores y diseñadores.
- Cambios de interfaz de usuario no requieren re compilación.
- Amigable a diseñadores CSS/Javascript.

Desventajas

- Es un lenguaje de programación bastante nuevo por lo que existe poca documentación.
- Es un paradigma de desarrollo distinto al acostumbrado (22).

1.7.2 C++

Es un lenguaje de programación muy utilizado mundialmente en la actualidad, surge en la década de los 80"s por parte de Bjarne Stroustrup, con el objetivo de mejorar el lenguaje de programación anteriormente publicado, el Lenguaje C, este es un lenguaje híbrido desde el punto de vista de los lenguajes orientados a objetos. Una de las mejores particularidades del lenguaje C++ es la de poder redefinir sus operadores. Su nombre viene a darse en 1983 por Rick Mascitti, el cual significa (C++) "incremento de C", refiriendo a este como una actualización al lenguaje C.

Algunas de sus características distintivas clave incluyen:

- Muy claro, sintaxis legible.
- La orientación a objetos intuitiva.
- Expresión natural del código de procedimiento.
- Extensas bibliotecas estándar y módulos de terceros para la mayoría de las tareas.
- Extensiones y módulos fácilmente escritos en C, C + + (o Java para Jython, o. NET para IronPython).
- Integrable dentro de las aplicaciones como una interfaz de scripting.
- Poderoso y rápido.
- Multiplataforma (Linux / Unix, Windows, OS / 2, Mac, entre otros).
- Fácil aprendizaje.
- Libre uso y distribuibles, incluso para uso comercial (23).

1.7.3 JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. En la actualidad es uno de los lenguajes de programación más usados debido a sus ventajas entre las cuales están: técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, en otras palabras, los programas escritos con JavaScript se pueden probar

directamente en cualquier navegador sin necesidad de procesos intermedios por lo que el tiempo de ejecución estará cerca de una aplicación creada en C o C++; cuenta con JSON (JavaScript Object Notation) que se utiliza en casi todas las aplicaciones web modernas, tanto para la comunicación, como para la persistencia de los datos; soporta características como la encapsulación, polimorfismo, la herencia múltiple y la composición; su sintaxis es inmediatamente familiar a cualquiera que tenga experiencia en C, C++, Java, PHP entre otros; casi todo en JavaScript es un objeto, incluyendo las funciones, debido a esa característica, las funciones se pueden utilizar en casi cualquier lugar y además el código JavaScript, en virtud de su ubicuidad, es el código reutilizable más portátil alrededor del mundo (24).

Se utilizará como lenguaje de programación C++, pues la biblioteca QtWebsocket es una de las implementaciones de la tecnología Websocket que está escrita en este lenguaje, por lo que es necesaria la utilización de C++ para la implementación de la comunicación de la aplicación. Además se utilizará QML para la creación de interfaces de usuario y JavaScript para el llenado de los modelos y validaciones en las interfaces, los mismos son utilizados ya que son los únicos lenguajes soportados por la plataforma de desarrollo escogida QtQuick.

1.8 Herramientas CASE

Las herramientas CASE (Ingeniería de Software Asistida por Computadora) son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. Constituyen un soporte automatizado para el desarrollo y mantenimiento del software. Se pueden ver como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales.

1.8.1 Visual Paradigm

Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones que utiliza modelado UML. Es ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos.

Ventajas del Visual Paradigm:

- Navegación intuitiva entre la escritura del código y su visualización

- Potente generador de informes en formato PDF/HTML
- Documentación automática Ad-hoc
- Ambiente visualmente superior de modelado
- Sofisticado diagramador automático.
- Sincronización de código fuente en tiempo real (25).

1.9 Metodología de Desarrollo de Software

Una Metodología de Desarrollo de Software es una filosofía, herramientas, modelos y métodos para asistir al proceso de desarrollo de software. Hasta hace poco este proceso de desarrollo llevaba asociada un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Este esquema tradicional ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos). Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad.

Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del “buen hacer” de la ingeniería del software, asumiendo el riesgo que ello significa (26). En este escenario, las metodologías ágiles emergen como una posible respuesta para llenar ese vacío metodológico. Por estar especialmente orientadas para proyectos pequeños, con plazos reducidos, requisitos volátiles, y/o basados en nuevas tecnologías.

1.9.1 Programación extrema (XP)

La programación extrema es una metodología de desarrollo ligero (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas. Este modelo de programación se basa en una serie de metodologías de desarrollo de software, brindándole prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación.

La Programación Extrema se basa en 12 principios básicos agrupados en cuatro categorías:

- **Retroalimentación a escala fina**

1. El principio de pruebas: se tiene que establecer un período de pruebas de aceptación del programa (llamado también período de caja negra) donde se definirán las entradas al sistema y los resultados esperados de estas entradas.

2. Proceso de planificación: en esta fase, el usuario tendrá que escribir sus necesidades, definiendo las actividades que realizará el sistema. Se creará un documento llamado Historias del usuario (User Stories).

3. El cliente en el sitio: se le dará poder para determinar los requerimientos, definir la funcionalidad, señalar las prioridades y responder las preguntas de los programadores.

4. Programación en parejas: uno de los principios más radicales y en el que la mayoría de gerentes de desarrollo ponen sus dudas.

- **Proceso continuo en lugar de por lotes**

1. Integración continua: permite al equipo hacer un rápido progreso implementando las nuevas características del software.

2. Refactorización: permite a los equipos de programadores XP mejorar el diseño del sistema a través de todo el proceso de desarrollo.

3. Entregas pequeñas: colocan un sistema sencillo en producción rápidamente, que se actualiza de forma ágil y constante permitiendo que el verdadero valor de negocio del producto sea evaluado en un ambiente real.

- **Entendimiento compartido**

1. Diseño simple: se basa en la filosofía de que el mayor valor de negocio es entregado por el programa más sencillo que cumpla los requerimientos.

2. Metáfora: desarrollada por los programadores al inicio del proyecto, define una historia de cómo funciona el sistema completo. XP estimula historias, que son breves descripciones de trabajo de un sistema en lugar de los tradicionales diagramas y modelos UML (Unified Modeling Language). La metáfora expresa la visión evolutiva del proyecto que define el alcance y propósito del sistema. Las tarjetas CRC (Clase, Responsabilidad y Colaboración) también ayudarán al equipo a definir actividades durante el diseño del sistema. Cada tarjeta

representa una clase en la programación orientada a objetos y define sus responsabilidades (lo que ha de hacer) y las colaboraciones con las otras clases (cómo se comunica con ellas).

3. Propiedad colectiva del código: un código con propiedad compartida. Nadie es el propietario de nada, todos son el propietario de todo.

4. Estándar de codificación: define la propiedad del código compartido así como las reglas para escribir y documentar el código y la comunicación entre distintas piezas de código desarrolladas por diferentes equipos.

- **Bienestar del programador.**

1. La semana de 40 horas: la programación extrema sostiene que los programadores cansados escriben código de menor calidad (27).

La metodología XP es seleccionada para guiar este trabajo de diploma pues el equipo de desarrollo es pequeño y se cuenta con poco tiempo para el diseño e implementación de la aplicación. Se identificó también que los requisitos de la misma son muy cambiantes, por lo cual una metodología más robusta como RUP (Rational Unified Process) retrasaría el trabajo pues no está concebida para este tipo de proyectos, mientras que XP es más flexible a este tipo de cambios. Además en la metodología XP todos tienen conocimiento del código en general y de que tareas están realizando los demás compañeros, esto es un aspecto muy deseable en un trabajo de diploma. Un aspecto clave a tener en cuenta es también la facilidad de aprendizaje de cada metodología y según estudios realizados previamente (28) XP es una metodología que tiene un nivel medio en cuanto facilidad de aprendizaje se refiere donde los miembros entienden bastante bien la metodología.

1.10 Consideraciones parciales

En este capítulo se detallaron las principales características de los sistemas SCADA y especialmente de uno de sus módulos el HMI, se explicaron sus características y funciones fundamentales así como algunos de los estándares que existen para su diseño. Además se realizó un estudio de la tecnología móvil en la actualidad, donde se profundizó en elementos como sus características, ventajas y los aspectos a tener en cuenta para desarrollar aplicaciones para este tipo de entorno.

Se realizó una descripción de las tecnologías escogidas para la implementación de la solución, las cuales son: protocolo WebSocket para la comunicación, la biblioteca

QTWebSocket que implementa dicho protocolo y como plataforma de desarrollo QtQuick. Como lenguajes de programación se escogió C++ ya que la biblioteca QTWebsocket está implementada en este lenguaje y QML y JavaScript para el trabajo con QTQuick, así como el IDE de desarrollo QT Creator debido a las facilidades que brinda para los lenguajes escogidos, garantizando la integración con el proyecto SCADA GALBA. Como herramienta de modelado se seleccionó Visual Paradigm, ya que se caracteriza por ser muy intuitiva. Como metodología de desarrollo se seleccionó XP por ser una metodología ágil, que propicia varias ventajas para proyectos pequeños.

2 Capítulo 2: Solución propuesta

2.1 Introducción

En el presente capítulo se interpretan las necesidades del sistema especificándolas mediante los requerimientos funcionales y no funcionales. Además se detallan brevemente los artefactos de la metodología de desarrollo XP, seleccionada para guiar el desarrollo del proyecto. Describiéndose las historias de usuario y las tareas de ingeniería asociadas a las mismas.

2.2 Propuesta del sistema

La solución que se propone abarca la creación de un cliente HMI para dispositivos móviles para el sistema SCADA Guardián del Alba. Este se ejecutará sobre dispositivos móviles que utilicen el sistema operativo Android, con posibilidades de extender la compatibilidad a iOS o BlackBerry y le permitirá al usuario visualizar las alarmas, las señales analógicas y digitales, la información de los eventos del sistema, los dispositivos y los subcanales, así como sus detalles asociados y gráficas de tendencias. Además permitirá conocer el estado de los módulos. La siguiente figura muestra gráficamente la solución propuesta:

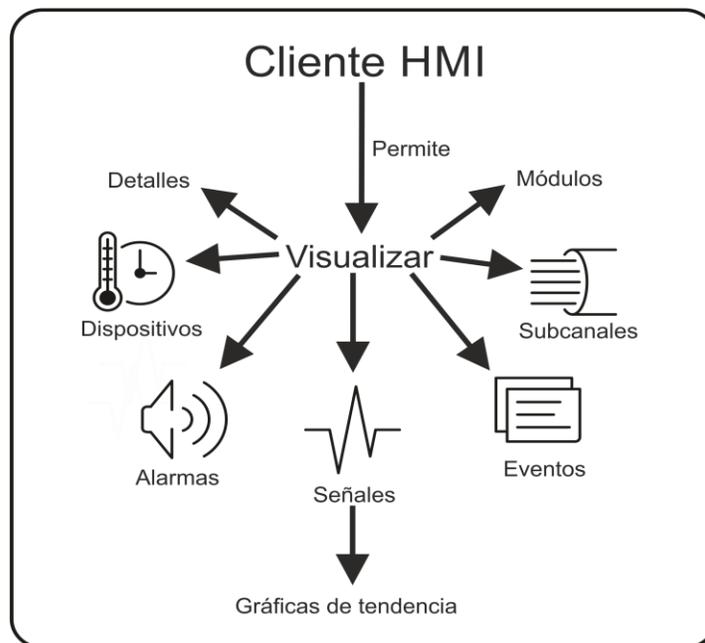


Figura 5: Propuesta de Sistema.

2.3 Fase de exploración y planificación

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Además 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. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente (29).

2.3.1 Historias de usuario

Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho énfasis en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas (30).

A continuación se muestran las tablas con las historias de usuario de producto que tiene como resultado la presente investigación.

Tabla 2: Historia de usuario: Autenticar Usuario.

| Historia de Usuario | |
|--|---|
| Número: HU_1 | Nombre de Historia de Usuario: Autenticar Usuario. |
| Actor: Usuario | Iteración asignada: 1 |
| Prioridad en negocio: Alta | Puntos estimados: 1 |
| Nivel de Complejidad: Alta | Puntos reales: 1 |
| Descripción: El sistema debe permitir autenticar un usuario con usuario y contraseña. | |
| Observaciones: | |

Tabla 3: Tarea: Conexión con el servidor.

| Tarea | |
|---|------------------------------|
| Número: 1 | Número Historia: 1 |
| Nombre de Tarea: Conexión con el servidor. | Iteración asignada: 1 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Descripción: Con esta tarea se le permite al usuario conectarse a un servidor especificado por él y desconectarse del mismo. | |
| Observaciones: | |

Tabla 4: Autenticarse en el servidor.

| Tarea | |
|---|------------------------------|
| Número: 2 | Número Historia: 1 |
| Nombre de Tarea: Autenticarse en el servidor | Iteración asignada: 1 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Descripción: Con esta tarea se le permite al usuario autenticarse en el servidor con usuario y contraseña. | |
| Observaciones: | |

Tabla 5: Historia de usuario: Mostrar sumario de alarmas.

| Historia de Usuario | |
|---|--|
| Número: HU_2 | Nombre de Historia de Usuario: Mostrar sumario de alarmas |
| Actor: Usuario | Iteración asignada: 1 |
| Prioridad en negocio: Alta | Puntos estimados: 2 |
| Nivel de Complejidad: Alta | Puntos reales: 2 |
| Descripción: El sistema debe mostrar al usuario un sumario de alarmas. | |
| Observaciones: | |

Tabla 6: Tarea: Implementar interfaz del sumario de Alarmas.

| Tarea | |
|---|------------------------------|
| Número: 1 | Número Historia: 2 |
| Nombre de Tarea: Implementar interfaz del sumario de Alarmas | Iteración asignada: 1 |

| | |
|---|------------------------------|
| Tipo de tarea: Desarrollo | Puntos estimados: 1.5 |
| Descripción: Con esta tarea se permite al usuario visualizar el sumario de alarmas y los detalles asociados a cada alarma. | |
| Observaciones: | |

Tabla 7: Tarea: Filtrar alarmas.

| Tarea | |
|--|------------------------------|
| Número: 2 | Número Historia: 2 |
| Nombre de Tarea: Filtrar alarmas | Iteración asignada: 1 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Descripción: Con esta tarea se permite al usuario filtrar las alarmas por la propiedad "recurso". | |
| Observaciones: | |

Tabla 8: Historia de Usuario: Mostrar sumario de señales analógicas.

| Historia de Usuario | |
|--|---|
| Número: HU_3 | Nombre de Historia de Usuario: Mostrar sumario de señales analógicas |
| Actor: Usuario | Iteración asignada: 2 |
| Prioridad en negocio: Alta | Puntos estimados: 2 |
| Nivel de Complejidad: Alta | Puntos reales: 2 |
| Descripción: El sistema muestra un sumario con las señales analógicas al usuario. | |
| Observaciones: | |

Tabla 9: Tarea: Implementar interfaz del sumario de señales analógicas.

| Tarea | |
|--|------------------------------|
| Número: 1 | Número Historia: 3 |
| Nombre de Tarea: Implementar interfaz del sumario de señales analógicas | Iteración asignada: 2 |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |

| |
|--|
| Descripción: Con esta tarea se permite al usuario visualizar el sumario de señales analógicas y los detalles generales y de mantenimiento asociados a cada una. |
| Observaciones: |

Tabla 10: Tarea: Filtrar señales analógicas.

| Tarea | |
|--|------------------------------|
| Número: 2 | Número Historia: 3 |
| Nombre de Tarea: Filtrar señales analógicas | Iteración asignada: 2 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Descripción: Con esta tarea se permite al usuario filtrar las señales por GOP (grupo operacional de privilegios), GTH (grupo de transferencia a históricos) y dispositivos. | |
| Observaciones: | |

Tabla 11: Tarea: Gráficas de tendencia para señales analógicas.

| Tarea | |
|--|------------------------------|
| Número: 3 | Número Historia: 3 |
| Nombre de Tarea: Gráficas de tendencia para señales analógicas. | Iteración asignada: 2 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Descripción: El sistema debe mostrar al usuario gráficas de tendencia con los valores en tiempo real de las señales analógicas. | |
| Observaciones: | |

Tabla 12: Historia de Usuario: Mostrar sumario de señales digitales.

| Historia de Usuario | |
|--|--|
| Número: HU_4 | Nombre de Historia de Usuario: Mostrar sumario de señales digitales |
| Actor: Usuario | Iteración asignada: 3 |
| Prioridad en negocio: Alta | Puntos estimados: 2 |
| Nivel de Complejidad: Alta | Puntos reales: 2 |
| Descripción: El sistema muestra al usuario el sumario de señales digitales. | |
| Observaciones: | |

Tabla 13: Tarea: Implementar interfaz del sumario de señales digitales.

| Tarea | |
|---|------------------------------|
| Número: 1 | Número Historia: 4 |
| Nombre de Tarea: Implementar interfaz del sumario de señales digitales | Iteración asignada: 3 |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Descripción: Con esta tarea se permite al usuario visualizar el sumario de señales digitales y los detalles generales y de mantenimiento asociados a cada una. | |
| Observaciones: | |

Tabla 14: Tarea: Filtrar señales digitales.

| Tarea | |
|--|------------------------------|
| Número: 2 | Número Historia: 4 |
| Nombre de Tarea: Filtrar señales digitales | Iteración asignada: 3 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Descripción: Con esta tarea se permite al usuario filtrar las señales por GOP (grupo operacional de privilegios), GTH (grupo de transferencia a históricos) y dispositivos. | |
| Observaciones: | |

Tabla 15: Tarea: Gráficas de tendencia para señales digitales.

| Tarea | |
|--|------------------------------|
| Número: 3 | Número Historia: 4 |
| Nombre de Tarea: Gráficas de tendencia para señales digitales. | Iteración asignada: 3 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.5 |
| Descripción: La aplicación debe mostrar al usuario gráficas de tendencia con los valores en tiempo real de las señales digitales. | |
| Observaciones: | |

Tabla 16: Historia de Usuario: Mostrar eventos.

| Historia de Usuario | |
|-----------------------|---|
| Número: HU_5 | Nombre de Historia de Usuario: Mostrar eventos |
| Actor: Usuario | Iteración asignada: 2 |

| | |
|---|----------------------------|
| Prioridad en negocio: Alta | Puntos estimados: 1 |
| Nivel de Complejidad: Alta | Puntos reales: 1 |
| Descripción: El sistema debe mostrar al usuario los eventos del sistema. | |
| Observaciones: | |

Tabla 17: Implementar interfaz de sumario de Eventos.

| Tarea | |
|--|-------------------------------|
| Número: 1 | Número Historia: 5 |
| Nombre de Tarea: Implementar interfaz de sumario de Eventos | Iteración asignada: 2 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.75 |
| Descripción: Con esta tarea se permite al usuario visualizar el sumario de eventos y los detalles asociados a cada uno. | |
| Observaciones: | |

Tabla 18: Tarea: Filtrar eventos.

| Tarea | |
|--|-------------------------------|
| Número: 1 | Número Historia: 5 |
| Nombre de Tarea: Filtrar eventos | Iteración asignada: 2 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.25 |
| Descripción: El sistema debe permitir al usuario filtrar los eventos del sistema basado en diferentes parámetros: comando, reconocimiento, retorno a normal, sesión, permiso, monitor, cambio en configuración. | |
| Observaciones: | |

Tabla 19: Historia de Usuario: Monitor del sistema.

| Historia de Usuario | |
|-----------------------------------|---|
| Número: HU_6 | Nombre de Historia de Usuario: Monitor del sistema |
| Actor: Usuario | Iteración asignada: 4 |
| Prioridad en negocio: Alta | Puntos estimados: 3 |
| Nivel de Complejidad: Alta | Puntos reales: 3 |

| |
|--|
| Descripción: El sistema debe permitir al usuario tener información sobre el estado de los módulos, subcanales y dispositivos. |
| Observaciones: |

Tabla 20: Tarea: Mostrar estado de los módulos.

| Tarea | |
|---|------------------------------|
| Número: 1 | Número Historia: 6 |
| Nombre de Tarea: Mostrar estado de los módulos | Iteración asignada: 4 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.4 |
| Descripción: Con esta tarea se debe permitir al usuario conocer el estado de los módulos seguridad, configuración, histórico, recolector, alarmas, comandos, monitor, puntos y publicador. | |
| Observaciones: | |

Tabla 21: Tarea: Implementar interfaz de sumario de subcanales.

| Tarea | |
|---|------------------------------|
| Número: 3 | Número Historia: 6 |
| Nombre de Tarea: Implementar interfaz de sumario de subcanales | Iteración asignada: 4 |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Descripción: Con esta tarea se debe permitir al usuario visualizar el sumario de subcanales y los detalles asociados a cada uno. | |
| Observaciones: | |

Tabla 22: Tarea: Filtrar subcanales.

| Tarea | |
|---|------------------------------|
| Número: 4 | Número Historia: 6 |
| Nombre de Tarea: Filtrar Subcanales | Iteración asignada: 4 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.3 |
| Descripción: Con esta tarea se debe permitir al usuario filtrar los subcanales por la propiedad "recurso". | |
| Observaciones: | |

Tabla 23: Tarea: Implementar interfaz de sumario de dispositivos.

| Tarea | |
|---|------------------------------|
| Número: 5 | Número Historia: 6 |
| Nombre de Tarea: Implementar interfaz de sumario de dispositivos. | Iteración asignada: 4 |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Descripción: Con esta tarea se debe permitir al usuario visualizar el sumario de dispositivos y los detalles asociados a cada uno. | |
| Observaciones: | |

Tabla 24 : Tarea: Filtrar dispositivos.

| Tarea | |
|--|------------------------------|
| Número: 6 | Número Historia: 6 |
| Nombre de Tarea: Filtrar dispositivos | Iteración asignada: 4 |
| Tipo de tarea: Desarrollo | Puntos estimados: 0.3 |
| Descripción: El sistema debe permitir al usuario filtrar los dispositivos por la propiedad "recurso". | |
| Observaciones: | |

Tabla 25: Historia de Usuario: Proveer ayuda.

| Historia de Usuario | |
|---|--|
| Número: HU_7 | Nombre de Historia de Usuario: Proveer ayuda. |
| Actor: Usuario | Iteración asignada: 3 |
| Prioridad en negocio: Alta | Puntos estimados: 1 |
| Nivel de Complejidad: Alta | Puntos reales: 1 |
| Descripción: El sistema debe brindar al usuario una ayuda para el uso de la aplicación, la cual debe contener una leyenda, una descripción de cómo usar la aplicación y proveer vínculos de interés institucional. | |
| Observaciones: | |

Actores

Tabla 26: Actores.

| Actores | Descripción |
|----------|---|
| Usuarios | Realiza peticiones al cliente de visualización. |

2.3.2 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable, por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser (31).

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con la toda la funcionalidad requerida, las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación (31).

RNF1 Software

La aplicación deberá ser ejecutada en un dispositivo móvil que utilice el sistema operativo Android a partir de su versión 2.3, con posibilidades en un futuro cercano de expandir la compatibilidad a iOS 5 o superior y BlackBerry OS 10 o superior.

RNF2 Hardware

La aplicación será utilizada en dispositivos móviles, los cuales disponen de pocos recursos, por lo que debe ejecutarse sin problemas en plataformas con un núcleo y con 128 mb de memoria RAM. El dispositivo en el que se ejecute deberá poseer pantalla táctil no importando el tamaño de la misma aunque se recomienda que sea mayor de 2,5 pulgadas.

RNF3 Apariencia o interfaz externa

La aplicación deberá tener una interfaz externa con un diseño moderno. También deberá regirse por los estándares internacionales establecidos para los HMI (5) (6). Se tendrá en cuenta también a la hora del diseño de las interfaces los retos del diseño para móviles.

RNF4 Seguridad

La aplicación no debe permitir ejercer ningún tipo de control sobre el proceso industrial, además en orden de acceder al servidor el usuario deberá autenticarse con un usuario y contraseña autorizados.

RNF5 Usabilidad

La aplicación debe proveer ayuda al usuario sobre el uso de la misma, posibilitándole acceder a una ayuda o leyenda sobre la aplicación. Se debe mostrar al usuario solo la información crítica para él, pero se debe dar la posibilidad de obtener más información en caso necesario.

2.3.3 Estimación de esfuerzo por historias de usuario

Las estimaciones de esfuerzo asociado a la implementación de las historias las establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

Tabla 27: Estimación de esfuerzo por historias de usuario.

| Historia de usuario | Esfuerzo en semanas |
|---------------------------------------|---------------------|
| Autenticar usuario | 1 |
| Mostrar sumario de alarmas | 2 |
| Mostrar sumario de señales analógicas | 2 |
| Mostrar sumario de señales digitales | 2 |
| Mostrar sumario de eventos | 1 |
| Monitor de sistema | 3 |
| Proveer ayuda | 1 |
| Total | 12 |

2.3.4 Plan de iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Iteraciones está compuesto por iteraciones de no más de tres semanas. 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 fueren 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 de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de Iteraciones son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior (32).

Tabla 28: Plan de iteraciones.

| Iteración | No.HU | Historia de usuario(HU) | Duración estimada (semanas) |
|-----------|-------|---------------------------------------|-----------------------------|
| 1 | HU_1 | Autenticar Usuario | 3 |
| | HU_2 | Mostrar sumario de alarmas | |
| 2 | HU_3 | Mostrar sumario de señales analógicas | 3 |
| | HU_5 | Mostrar eventos | |
| 3 | HU_4 | Mostrar sumario de señales digitales | 3 |
| | HU_7 | Proveer ayuda | |
| 4 | HU_6 | Monitor de sistema | 3 |

2.3.5 Plan de entregas

Las historias de usuario servirán para crear el plan estimado de entrega. El plan de entregas se usará para crear los planes de iteración para cada iteración. Es en este momento cuando clientes y desarrolladores toman las decisiones comerciales y técnicas respectivamente. Con cada historia de usuario previamente evaluada en tiempo de desarrollo ideal, el cliente las agrupará en orden de importancia. De esta forma se puede trazar el plan de entregas en función de estos dos parámetros: tiempo de desarrollo ideal y grado de importancia para el cliente.

Este artefacto se elabora con la intención de fijar qué período de tiempo puede tardar la implementación de cada una de las historias, definiéndose las fechas en que serán liberadas las versiones funcionales del producto (32).

Tabla 29: Plan de entregas.

| Iteración | Iteración No. 1 | Iteración No. 2 | Iteración No. 3 | Iteración No. 4 |
|---|-----------------------|------------------|---------------------|-------------------|
| Cantidad de historias de usuario | 2 | 2 | 2 | 1 |
| Fechas de entrega | 28 de Febrero de 2014 | 21 Marzo de 2014 | 11 de Abril de 2014 | 2 de Mayo de 2014 |

2.4 Diseño de la solución propuesta

El diseño de la solución propuesta se llevará a cabo teniendo en cuenta el proceso de visualización de sumarios del ambiente de escritorio del SCADA Guardián del ALBA. Diseñándose una aplicación que se ejecute sobre un dispositivo móvil y que permita mediante la conexión con el servidor mostrar al usuario la información requerida por este. Para dicha conexión se utilizará el protocolo de comunicación QtWebsocket. Los detalles de la aplicación se podrán apreciar mejor a través de su arquitectura.

2.4.1 Arquitectura

La arquitectura es la organización fundamental de un sistema materializándose en sus componentes, las relaciones entre ellos, el ambiente y los principios guiando su diseño y evolución (33). En la aplicación se identifica la necesidad de utilización de 2 patrones arquitectónicos Cliente-servidor y Modelo-Vista. A continuación se explicará con más detalle cada uno de ellos y donde se reflejan en el sistema desarrollado.

Cliente servidor

La utilización de este patrón posibilita que se mantenga una clara separación entre la lógica de negocio, la presentación y el acceso a datos. Permitiendo flexibilidad y facilidad a la hora de realizar posibles modificaciones (34). La siguiente figura muestra un diagrama general de esta arquitectura:

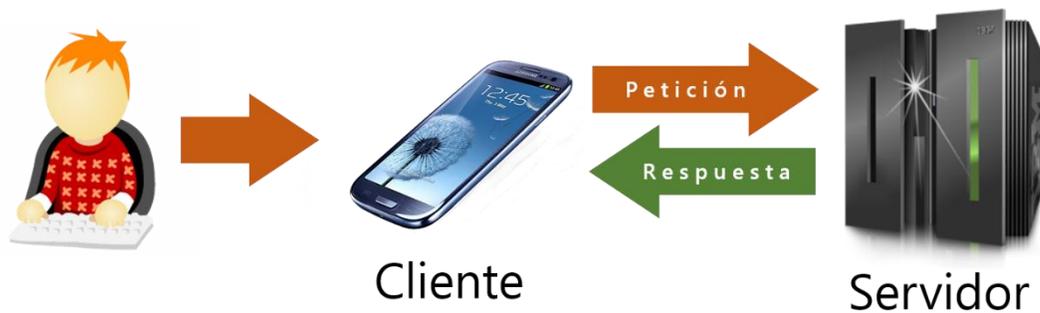


Figura 6: Patrón Cliente-Servidor.

Este patrón se ve reflejado en la aplicación a la hora de la obtención de los datos, pues cada vez que se requiere mostrar información al usuario se hace una petición al servidor con la información deseada, este interpreta la petición y envía una respuesta al cliente, y este muestra la información al usuario. La siguiente figura ilustra lo previamente expuesto:

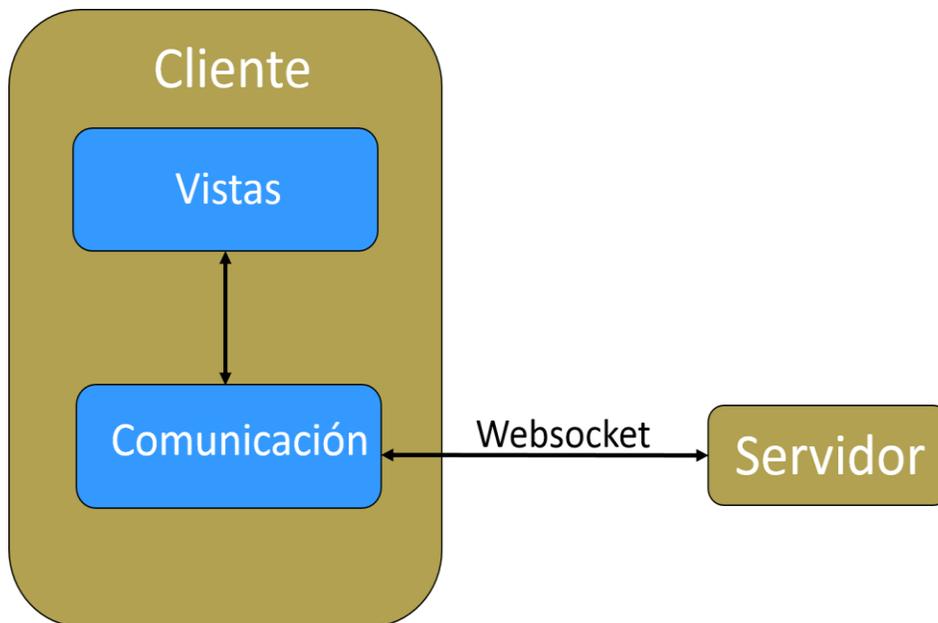


Figura 7: Patrón Cliente-Servidor en la aplicación.

Comunicación: Implementa la comunicación con el servidor en el lenguaje C++. Esta se encargará del envío de las peticiones realizadas en las vistas mediante JSON al servidor y de interpretar la respuesta del mismo, enviando a las vistas los datos en un formato que sea más entendible para el mismo.

Vistas: Estas son el centro de la aplicación, engloban un lenguaje declarativo que se encarga de la creación de los elementos visuales el QML y el lenguaje JavaScript que se encarga de la lógica del negocio. Además se nutre también de la conexión con el servidor para responder a las acciones del usuario creando una armónica simbiosis entre todos los elementos del sistema.

Modelo/Vista

Modelo-vista-controlador es un patrón de diseño que se originó de SmallTalk y que es usado comúnmente al construir interfaces de usuario. Si la vista y el controlador son combinados el resultado es la arquitectura modelo/vista. Esta también separa la forma en que la información es almacenada de la forma en que se presenta al usuario, pero provee un marco de trabajo más simple basado en los mismos principios. Esta separación permite presentar la misma información en diferentes vistas e implementar nuevos tipos de vistas sin cambiar las estructuras subyacentes.

El modelo se comunica con la fuente de datos, brindando una interfaz para los otros componentes de la arquitectura. La naturaleza de la comunicación depende del tipo de fuente de datos y de la forma en que es implementado el modelo. La vista obtiene índices del modelo, estos se refieren a objetos de datos. Al proveer estos índices al modelo la vista puede obtener la información de la fuente. En una vista normal, un delegado edita el objeto de datos, cuando este es editado el delegado se comunica directamente con el modelo usando los índices del modelo (35). De manera general esta arquitectura se representa de la siguiente forma:

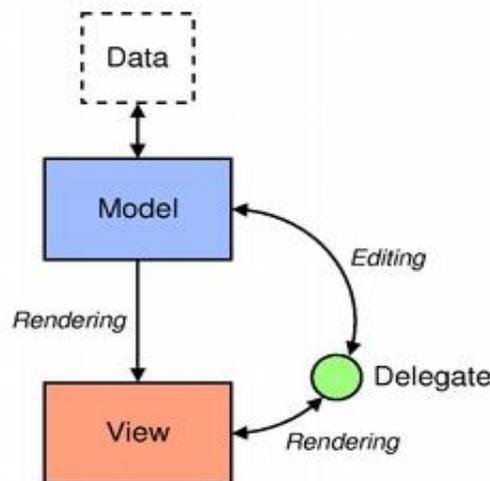


Figura 8: Patrón Modelo-Vista. Fuente: qt-project.org.

En la aplicación esta arquitectura se utiliza en los sumarios mostrados al usuario, presentándose de manera muy similar en cada uno de ellos. La siguiente figura muestra un diagrama de cómo se aplica esta arquitectura en las interfaces desarrolladas, usando como ejemplo la interfaz que despliega el sumario de alarmas:

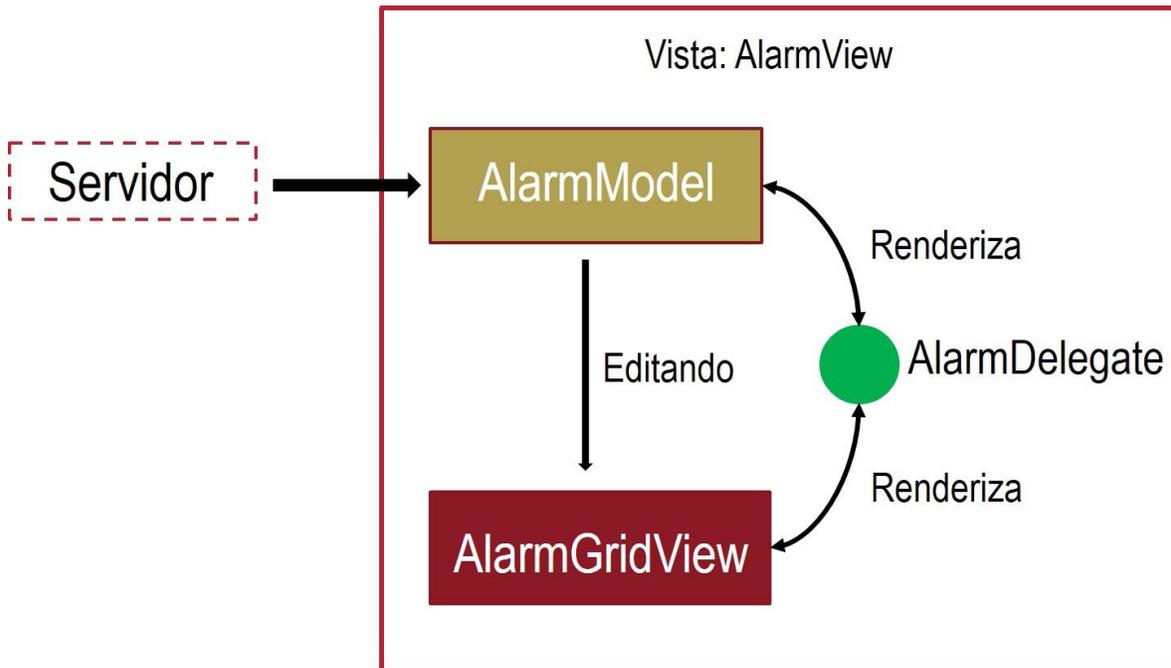


Figura 9: Patrón Modelo-Vista en la aplicación.

2.4.2 Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se debe tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios).

Patrones GRASP

GRASP es un acrónimo de General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para sugerir la importancia de aprehender (grasping en inglés) estos principios para diseñar con éxito el software orientado a objetos. Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. Es importante entender

y ser capaces de aplicar estos principios durante la creación de los diagramas de interacción porque un desarrollador de software con poca experiencia en la tecnología de objetos necesita dominar estos principios tan rápido como sea posible; constituyen la base de cómo se diseñará el sistema (36).

Patrón Controlador

El patrón controlador se encarga de asignar la responsabilidad de manejar los mensajes y eventos del sistema a una clase. En el sistema propuesto se ve este patrón en la clase Conexion que es quien controla todos los eventos del sistema, enviando las peticiones al servidor e interpretando las respuestas de este. En la siguiente figura se muestra como la clase Conexion maneja los eventos del sistema respondiendo a una petición del sumario de alarmas.

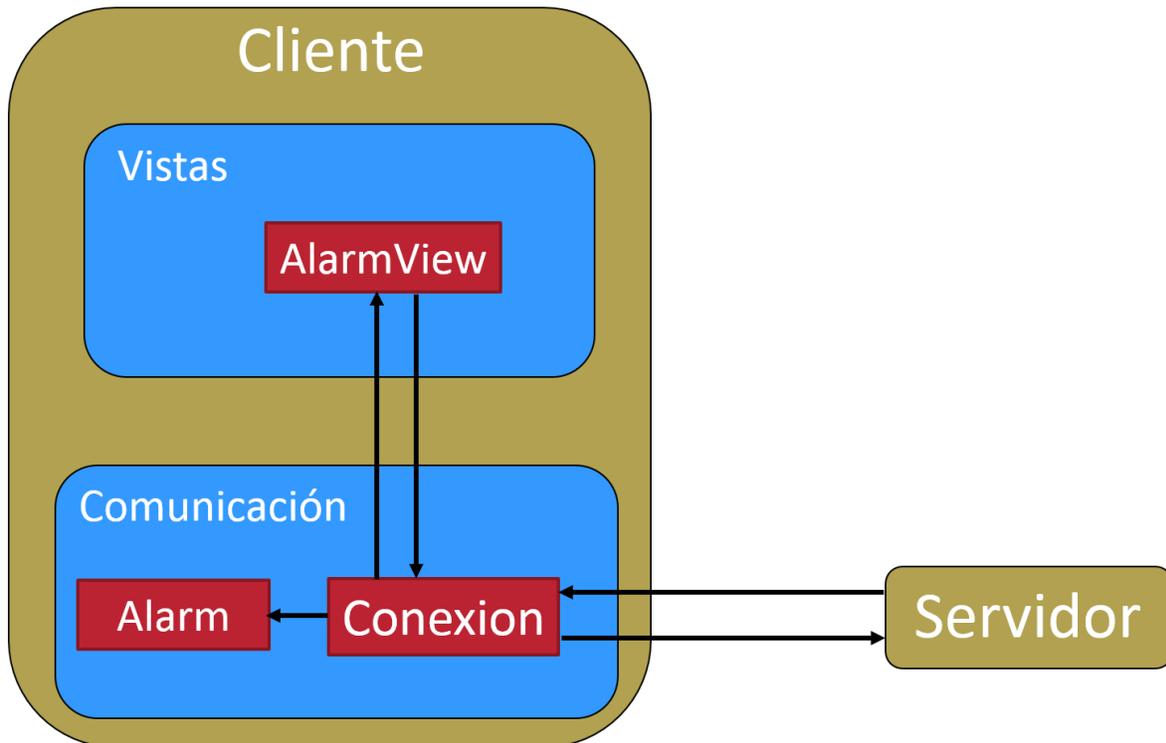


Figura 10: Patrón Controlador en la aplicación.

Patrón Creador

Asignar a la clase B la responsabilidad de crear una instancia de clase A si se cumple uno o más de los casos siguientes:

1. B agrega objetos de A.

2. B contiene objetos de A.
3. B registra instancias de objetos de A.
4. B utiliza más estrechamente objetos de A.
5. B tiene datos de inicialización que se pasarán a un objeto de A cuando sea creado (por tanto, B es un Experto con respecto a la creación de A).
6. B es un creador de los objetos A.

En la aplicación se ve este patrón aplicado en la clase Contenedor que contiene objetos de las interfaces, además esta clase es la que inicializa los objetos de las interfaces mediante una petición al servidor. En la siguiente figura se ilustra cómo la clase contenedor crea objetos de las demás interfaces del sistema:

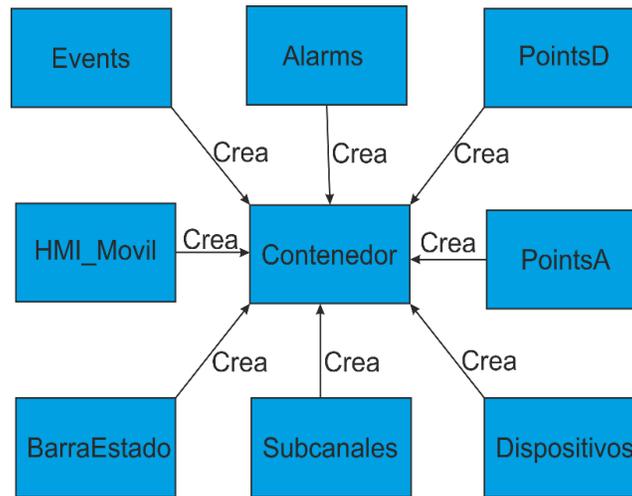


Figura 11: Patrón Creador en la aplicación.

2.4.3 Diagrama de clases

En el diagrama de clases se muestran los atributos, métodos y relaciones de las clases en el nivel de jerarquía más alto en el que son aplicables. Las relaciones existentes entre las distintas clases indican cómo se comunican los objetos de esas clases entre sí. Los mensajes de las diferentes clases navegan por las relaciones existentes entre las distintas clases (37). En la figura que se muestra a continuación se observa el diagrama de clases del sistema desarrollado:

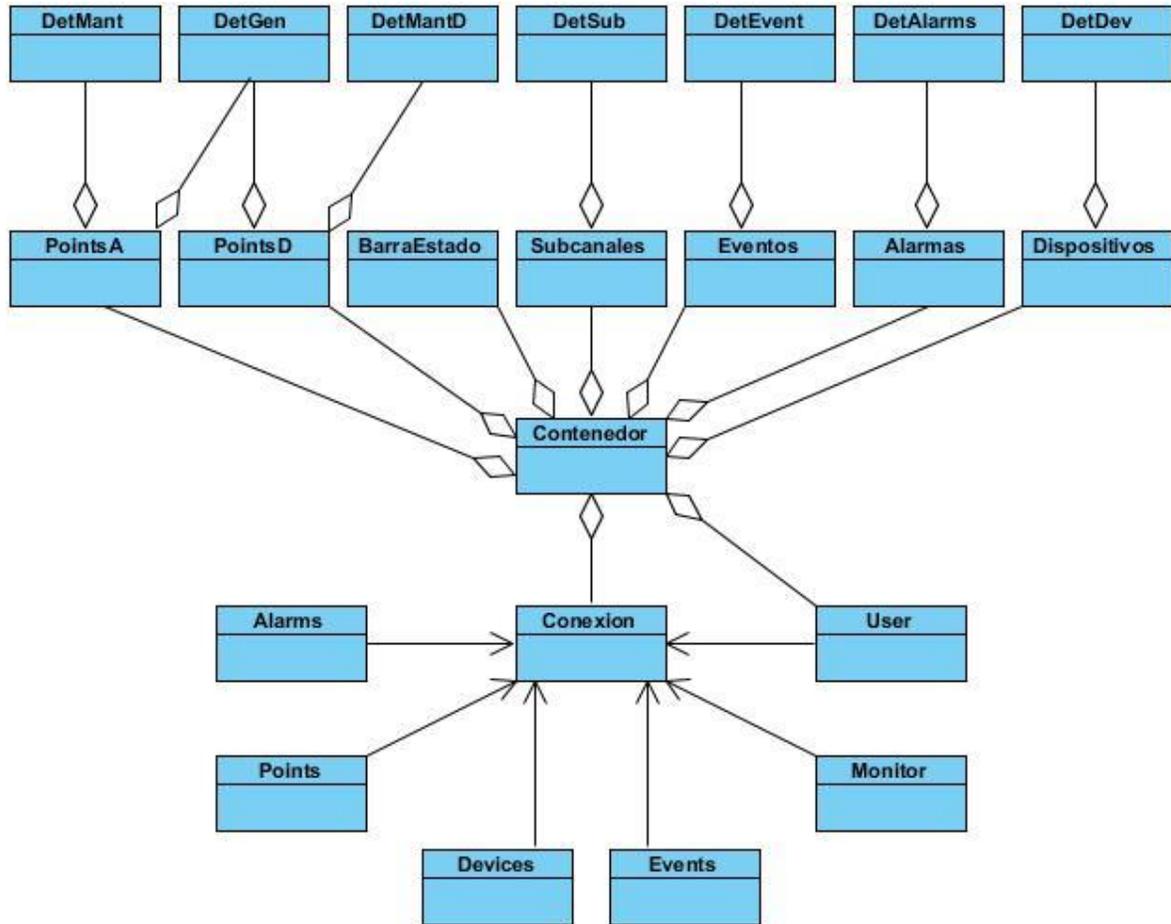


Figura 12: Diagrama de clases.

2.4.4 Tarjetas CRC

La utilización de tarjetas CRC siglas en inglés de Class-Responsibility-Collaboration es una técnica de diseño orientado a objetos, el objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que se necesitarán para implementar el sistema y la forma en que van a interactuar. De esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto, con el objeto de que el diseño sea lo más simple posible, verificando las especificaciones del sistema.

Tabla 30: Tarjeta CRC para la clase Conexion.

| |
|--|
| Clase: Conexion |
| Atributos: QWsSocket * wsSocket |
| Responsabilidad: Controlar la conexión con el servidor, recibiendo las respuestas de este y enviando al cliente de visualización los datos recibidos. |
| Colaboradores: QWsSocket, Alarms, User, Device, PointsScreen, QObject, QtGui/QGuiApplication, QtQuick/QQuickItem, QtQuick/QQuickView, QString, QJsonDocument, QJsonObject, QVariant, QJsonArray, QJsonValue |

Tabla 31: Tarjeta CRC para la clase Alarms.

| |
|---|
| Clase: Alarms |
| Atributos: QString help; QString timestamp; QString resource; QString cause; QString type; QString description; int occurrences; int priority; QString group; int behavior |
| Responsabilidad: Crear objetos de tipo alarmas. |
| Colaboradores: QObject, QtGui/QGuiApplication, QtQuick/QQuickItem, QtQuick/QQuickView |

Tabla 32: Tarjeta CRC para la clase Points.

| |
|---|
| Clase: Points |
| Atributos: QString id; QString name; QString description; QString engunit; QString group; QString qualitysymbol; QString type; int behavior; QString timestamp; QString deviceAliasName; QString channelAliasName; QString subChannelAliasName; QString vQualityDescription; QString inputAddress; QString outputAddress; QString ingeneerUnits; QString dataType; QString sampleFrequency; QString minIU; QString maxIU; QString deadBandPct; QString gthName; int value; int minField; int maxField; |
| Responsabilidad: Crear objetos de tipo punto. |
| Colaboradores: QObject, QString |

Tabla 33: Tarjeta CRC para la clase User.

| |
|---|
| Clase: User |
| Atributos: QString id; QString userName; qlonglong remainingTime |
| Responsabilidad: Crear objetos de tipo usuario. |
| Colaboradores: QObjects,QString |

Tabla 34: Tarjeta CRC para la clase Device.

| |
|--|
| Clase: Device |
| Atributos: int id; QString resource; QString description; QString actualstate; QString currentpolltime; QString previousstate; QString previouspolltime; QString name; QString timestamp; QString gopName; QString value; QString ingeneerUnits; int behavior |
| Responsabilidad: Crear objetos de tipo dispositivo. |
| Colaboradores: QObjects. |

Tabla 35: Tarjeta CRC para la clase Events.

| |
|--|
| Clase: Events |
| Atributos: QString timestamp; QString resourcename; QString eventtype; QString currentvalue; QString previouspointvalue; QString eventdescription; QString username; int behavior |
| Responsabilidad: Crear objetos de tipo evento. |
| Colaboradores: QObjects. |

Tabla 36: Tarjeta CRC para la clase Monitor.

| |
|--|
| Clase: Monitor |
| Atributos: int security; int historic; int alarms; int points; int publisher; int commands; int collector; int monitor; int configuration |
| Responsabilidad: Crear objetos de tipo monitor. |

Colaboradores: QObjects.

2.4.5 Diagrama de despliegue

Un diagrama de despliegue muestra la disposición física de los distintos nodos que componen un sistema y se indican cuáles son los enlaces de comunicación existentes entre los distintos componentes en tiempo de ejecución. En la siguiente figura se presenta el diagrama de despliegue del sistema desarrollado:

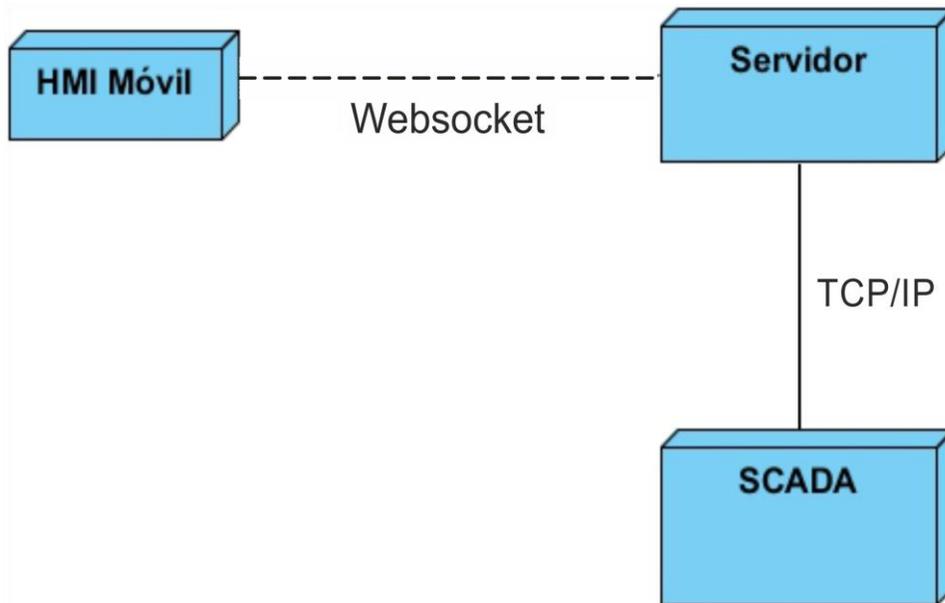


Figura 13: Diagrama de despliegue.

3 Capítulo 3: Validación de la propuesta

3.1 Introducción

Este capítulo centrará su atención en la realización de la validación y verificación del cumplimiento de los requerimientos planteados en el capítulo anterior con la aplicación de métodos de pruebas. Comprobando de esta forma el correcto funcionamiento de la aplicación basado en los objetivos expuestos por el cliente al inicio del proceso de desarrollo de esta.

3.2 Fase de pruebas.

Para la fase de producción el equipo se centró en los pedidos del cliente, refinando el diseño y el código continuamente. El perfeccionamiento de código sólo fue posible a través de un grupo de pruebas automatizadas que aseguraron la ejecución correcta del sistema en todo el período de desarrollo. Frecuentemente cuando se implementa, siempre se realizan una gran cantidad de pruebas para verificar que el código esté correcto. Estas pruebas normalmente tienen que ser realizadas varias veces y se ven afectadas por los cambios que se introducen conforme se va desarrollando.

Con el objetivo de evaluar la calidad de la aplicación se realizó un proceso de pruebas para validar la implementación de las funcionalidades definidas, efectuando pruebas aceptación.

3.2.1 Pruebas de aceptación

Las pruebas de aceptación comprueban las historias de usuario. Estas no solo le permiten asegurarse de compilar correctamente lo que el cliente necesita a lo largo del ciclo de vida del proyecto, sino también de consolidar la confianza del cliente y de mostrar las responsabilidades que ha aceptado. A continuación se muestran los casos de prueba definidos para la aplicación desarrollada.

Tabla 37: Tarjeta de aceptación "Conectar con el servidor".

| |
|---|
| Caso de prueba: 1 |
| No. Historia de usuario: 1 |
| Nombre de Caso de prueba: Conectar al servidor. |
| Descripción: Se realiza con el objetivo de comprobar que la aplicación se conecte correctamente al servidor. |

| |
|---|
| Condiciones de ejecución: El servidor debe encontrarse activo. |
| Entradas: Se oprime el botón Servidor, se introduce la dirección del servidor deseado y se oprime el botón Conectar. |
| Resultado esperado: <ul style="list-style-type: none"> • En caso de no ser correcta la dirección se muestra un mensaje de error con el siguiente texto: "Servidor no encontrado". • En caso contrario se realiza la conexión y se vuelve a la interfaz de autenticación. |
| Evaluación: Prueba exitosa. |

Tabla 38: Tarjeta de aceptación "Autenticar usuario".

| |
|--|
| Caso de prueba: 3 |
| No. Historia de usuario: 1 |
| Nombre de Caso de prueba: Autenticar usuario. |
| Descripción: Se realiza con el objetivo de comprobar que la funcionalidad autenticar usuario funcione correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de seguridad debe estar funcionando. |
| Entradas: Se introduce un usuario y contraseña válidos y se presiona el botón autenticar. |
| Resultado esperado: El sistema mostrará el mensaje de autenticación (error o éxito). En caso de ser exitosa la autenticación permitirá al usuario el acceso a la interfaz principal de la aplicación. |
| Evaluación: Prueba exitosa. |

Tabla 39: Tarjeta de aceptación "Cerrar sesión".

| |
|---|
| Caso de prueba: 4 |
| No. Historia de usuario: 1 |
| Nombre de Caso de prueba: Cerrar sesión. |
| Descripción: Se realiza con el objetivo de comprobar que la funcionalidad cerrar sesión funcione correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de seguridad debe estar funcionando. |
| Entradas: Se despliega la barra de estados que se encuentra en la parte superior de la aplicación, en esta se presiona el botón cerrar sesión. |

| |
|---|
| Resultado esperado: El sistema desconecta al usuario y muestra la interfaz de autenticación. |
| Evaluación: Prueba exitosa. |

Tabla 40: Tarjeta de aceptación "Mostrar sumario de alarmas".

| |
|--|
| Caso de prueba:1 |
| No. Historia de usuario: 2 |
| Nombre de Caso de prueba: Mostrar sumario de alarmas. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de alarmas se muestre correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón alarmas en la interfaz principal o en otro sumario. |
| Resultado esperado: <ul style="list-style-type: none"> • Si existen alarmas registradas en el sistema se muestra un sumario con las alarmas del sistema. • Si no existen alarmas en el sistema lanza un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". • Si el módulo de adquisición no está disponible el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| Evaluación: Prueba exitosa. |

Tabla 41: Tarjeta de aceptación "Filtrar alarmas"

| |
|---|
| Caso de prueba:2 |
| No. Historia de usuario: 2 |
| Nombre de Caso de prueba: Filtrar alarmas. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el filtrado de alarmas se haga correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón opciones de la interfaz del sumario de alarmas y oprime el botón filtrar de la barra lateral. Introduce el nombre del recurso por el que desea filtrar y presiona el botón filtrar. |

| |
|---|
| <p>Resultado esperado:</p> <ul style="list-style-type: none"> • Si existen alarmas que coincidan con el filtro especificado el sistema muestra al usuario solo las alarmas del recurso especificado. • En caso de no existir alarmas que coincidan con el filtro el sistema muestra un mensaje de error con el siguiente texto: “No hay mediciones que coincidan con el filtro”. • Si el módulo de adquisición está fuera de servicio el sistema lanza un mensaje de error con el siguiente texto: “No se puede conectar con el módulo de adquisición”. |
| <p>Evaluación: Prueba exitosa.</p> |

Tabla 42: Tarjeta de aceptación "Mostrar detalles de alarma".

| |
|---|
| Caso de prueba: 3 |
| No. Historia de usuario: 2 |
| Nombre de Caso de prueba: Mostrar detalles de alarmas. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se muestran los detalles de la alarma seleccionada. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona una alarma del sumario. |
| <p>Resultado esperado:</p> <ul style="list-style-type: none"> • Se muestran los detalles de la alarma. |
| Evaluación: Prueba exitosa. |

Tabla 43: Tarjeta de aceptación "Mostrar señales analógicas".

| |
|--|
| Caso de prueba: 1 |
| No. Historia de usuario: 3 |
| Nombre de Caso de prueba: Mostrar señales analógicas. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de señales analógicas se muestre correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario accede al sumario de señales analógicas desde la interfaz principal o desde otro sumario. |

| |
|--|
| <p>Resultado esperado:</p> <ul style="list-style-type: none"> • Si existen señales analógicas registradas en el sistema se muestra un sumario con las señales analógicas del sistema. • Si no existen señales analógicas en el sistema lanza un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". • Si el módulo de adquisición no está disponible el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| <p>Evaluación: Prueba exitosa.</p> |

Tabla 44: Tarjeta de aceptación "Filtrar señales analógicas".

| |
|---|
| <p>Caso de prueba:2</p> |
| <p>No. Historia de usuario: 3</p> |
| <p>Nombre de Caso de prueba: Filtrar señales analógicas.</p> |
| <p>Descripción: Esta prueba se realiza con el objetivo de comprobar que las señales analógicas se filtran correctamente.</p> |
| <p>Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo.</p> |
| <p>Entradas: El usuario presiona el botón opciones de la interfaz de señales analógicas, luego oprime el botón filtrar. Selecciona uno o varios parámetros a filtrar, introduce los valores de dichos parámetros y presiona el botón filtrar.</p> |
| <p>Resultado esperado:</p> <ul style="list-style-type: none"> • Si existen señales analógicas que coincidan con el filtro especificado, el sistema muestra al usuario solo las señales analógicas del recurso especificado. • En caso de no existir señales analógicas que coincidan con el filtro el sistema muestra un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". • Si el módulo de adquisición está fuera de servicio el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| <p>Evaluación: Prueba exitosa.</p> |

Tabla 45: Tarjeta de aceptación "Mostrar detalles de señales analógicas".

| |
|---|
| <p>Caso de prueba:3</p> |
| <p>No. Historia de usuario: 3</p> |
| <p>Nombre de Caso de prueba: Mostrar detalles de señales analógicas.</p> |

| |
|--|
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se muestran los detalles de la señal analógica seleccionada. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona una señal del sumario. |
| Resultado esperado: <ul style="list-style-type: none"> • Se muestran los detalles de la señal analógica. |
| Evaluación: Prueba exitosa. |

Tabla 46: Tarjeta de aceptación "Mostrar señales digitales".

| |
|--|
| Caso de prueba:1 |
| No. Historia de usuario: 4 |
| Nombre de Caso de prueba: Mostrar señales digitales. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de señales digitales se muestre correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario accede al sumario de señales digitales. Desde la interfaz principal o desde otro sumario. |
| Resultado esperado: <ul style="list-style-type: none"> • Si existen señales digitales registradas en el sistema se muestra un sumario con las señales digitales del sistema. • Si no existen señales digitales en el sistema lanza un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". • Si el módulo de adquisición no está disponible el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| Evaluación: Prueba exitosa. |

Tabla 47: Tarjeta de aceptación "Filtrar señales digitales".

| |
|---|
| Caso de prueba:2 |
| No. Historia de usuario: 4 |
| Nombre de Caso de prueba: Filtrar señales digitales. |

| |
|--|
| Descripción: Esta prueba se realiza con el objetivo de comprobar que las señales digitales se filtran correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón opciones de la interfaz de señales digitales, luego oprime el botón filtrar. Selecciona uno o varios parámetros a filtrar, introduce los valores de dichos parámetros y presiona el botón filtrar. |
| Resultado esperado: <ul style="list-style-type: none"> • Si existen señales digitales que coincidan con el filtro especificado el sistema muestra al usuario solo las señales digitales del recurso especificado. • En caso de no existir señales digitales que coincidan con el filtro el sistema muestra un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". • Si el módulo de adquisición está fuera de servicio el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| Evaluación: Prueba exitosa. |

Tabla 48: Tarjeta de aceptación "Mostrar detalles de señales digitales"

| |
|--|
| Caso de prueba:3 |
| No. Historia de usuario: 4 |
| Nombre de Caso de prueba: Mostrar detalles de señales digitales. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se muestran los detalles de la señal digital seleccionada. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona una señal del sumario. |
| Resultado esperado: <ul style="list-style-type: none"> • Se muestran los detalles de la señal digital. |
| Evaluación: Prueba exitosa. |

Tabla 49: Tarjeta de aceptación "Mostrar sumario de eventos".

| |
|-----------------------------------|
| Caso de prueba:1 |
| No. Historia de usuario: 5 |

| |
|--|
| Nombre de Caso de prueba: Mostrar sumario de eventos. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de eventos se muestre correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón eventos en la interfaz principal o en otro sumario. |
| Resultado esperado: <ul style="list-style-type: none"> • Si existen eventos registrados en el sistema se muestra un sumario con los eventos del sistema. • Si no existen eventos en el sistema lanza un mensaje de error con el siguiente texto: “No hay mediciones que coincidan con el filtro”. • Si el módulo de adquisición no está disponible el sistema lanza un mensaje de error con el siguiente texto: “No se puede conectar con el módulo de adquisición”. |
| Evaluación: Prueba exitosa. |

Tabla 50: Tarjeta de aceptación "Filtrar eventos".

| |
|--|
| Caso de prueba:2 |
| No. Historia de usuario: 5 |
| Nombre de Caso de prueba: Filtrar eventos. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de eventos se muestre correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón opciones de la interfaz del sumario de eventos y oprime el botón filtrar de la barra lateral. Selecciona el o los tipos de eventos por los que desea filtrar. |
| Resultado esperado: <ul style="list-style-type: none"> • Si existen eventos que coincidan con el filtro especificado el sistema muestra al usuario solo los tipos de eventos seleccionados. • En caso de no existir eventos que coincidan con el filtro el sistema muestra un mensaje de error con el siguiente texto: “No hay mediciones que coincidan con el filtro”. • Si el módulo de adquisición está fuera de servicio el sistema lanza un mensaje de error con el siguiente texto: “No se puede conectar con el módulo de adquisición”. |

| |
|------------------------------------|
| Evaluación: Prueba exitosa. |
|------------------------------------|

Tabla 51: Tarjeta de aceptación "Mostrar detalles de eventos".

| |
|---|
| Caso de prueba:3 |
| No. Historia de usuario: 5 |
| Nombre de Caso de prueba: Mostrar detalles de eventos. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se muestran los detalles del evento seleccionado. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona un evento del sumario. |
| Resultado esperado: <ul style="list-style-type: none"> Se muestran los detalles del evento. |
| Evaluación: Prueba exitosa. |

Tabla 52: Tarjeta de aceptación "Mostrar sumario de subcanales".

| |
|---|
| Caso de prueba:1 |
| No. Historia de usuario: 6 |
| Nombre de Caso de prueba: Mostrar sumario de subcanales. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de subcanales se muestre correctamente |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón subcanales en la interfaz principal o en otro sumario. |
| Resultado esperado: <ul style="list-style-type: none"> Si existen subcanales registrados en el sistema se muestra un sumario con los subcanales del sistema. Si no existen subcanales en el sistema lanza un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". Si el módulo de adquisición no está disponible el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| Evaluación: Prueba exitosa. |

Tabla 53: Tarjeta de aceptación "Filtrar subcanales".

| |
|---|
| Caso de prueba:2 |
| No. Historia de usuario: 6 |
| Nombre de Caso de prueba: Filtrar subcanales. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de subcanales se muestre correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón opciones de la interfaz del sumario de subcanales y oprime el botón filtrar de la barra lateral. Introduce el nombre del recurso por el que desea filtrar y presiona el botón filtrar. |
| Resultado esperado: <ul style="list-style-type: none"> • Si existen subcanales que coincidan con el filtro especificado el sistema muestra al usuario solo los subcanales que coincidan con dicho filtro. • En caso de no existir subcanales que coincidan con el filtro el sistema muestra un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". • Si el módulo de adquisición está fuera de servicio el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| Evaluación: Prueba exitosa. |

Tabla 54: Tarjeta de aceptación "Mostrar detalles de subcanal".

| |
|---|
| Caso de prueba:3 |
| No. Historia de usuario: 6 |
| Nombre de Caso de prueba: Mostrar detalles de subcanal. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se muestran los detalles del subcanal seleccionado. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona un subcanal del sumario. |
| Resultado esperado: <ul style="list-style-type: none"> • Se muestran los detalles del subcanal. |
| Evaluación: Prueba exitosa. |

Tabla 55: Tarjeta de aceptación "Mostrar sumario de dispositivos".

| |
|--|
| Caso de prueba:4 |
| No. Historia de usuario: 6 |
| Nombre de Caso de prueba: Mostrar sumario de dispositivos. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de dispositivos se muestre correctamente |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón dispositivos en la interfaz principal o en otro sumario. |
| <p>Resultado esperado:</p> <ul style="list-style-type: none"> • Si existen dispositivos registrados en el sistema se muestra un sumario con los dispositivos del sistema. • Si no existen dispositivos en el sistema lanza un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". <p>Si el módulo de adquisición no está disponible el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición".</p> |
| Evaluación: Prueba exitosa. |

Tabla 56: Tarjeta de aceptación "Filtrar dispositivos".

| |
|--|
| Caso de prueba:5 |
| No. Historia de usuario: 6 |
| Nombre de Caso de prueba: Filtrar dispositivos. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sumario de dispositivos se muestre correctamente. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona el botón opciones de la interfaz del sumario de dispositivos y oprime el botón filtrar de la barra lateral. Introduce el nombre del recurso por el que desea filtrar y presiona el botón filtrar. |
| <p>Resultado esperado:</p> <ul style="list-style-type: none"> • Si existen dispositivos que coincidan con el filtro especificado el sistema muestra al usuario solo los dispositivos que coincidan con dicho filtro. |

| |
|--|
| <ul style="list-style-type: none"> • En caso de no existir dispositivos que coincidan con el filtro el sistema muestra un mensaje de error con el siguiente texto: "No hay mediciones que coincidan con el filtro". • Si el módulo de adquisición está fuera de servicio el sistema lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| Evaluación: Prueba exitosa. |

Tabla 57: Tarjeta de aceptación "Mostrar detalles de dispositivos".

| |
|--|
| Caso de prueba:6 |
| No. Historia de usuario: 6 |
| Nombre de Caso de prueba: Mostrar detalles de dispositivo. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que se muestran los detalles del dispositivo seleccionado. |
| Condiciones de ejecución: Se debe estar conectado al servidor. El módulo de adquisición debe estar activo. |
| Entradas: El usuario presiona un dispositivo del sumario. |
| Resultado esperado: <ul style="list-style-type: none"> • Si el dispositivo se encuentra en el sistema se muestran los detalles del dispositivo. • Si el módulo de adquisición está fuera de servicio lanza un mensaje de error con el siguiente texto: "No se puede conectar con el módulo de adquisición". |
| Evaluación: Prueba exitosa. |

Tabla 58: Tarjeta de aceptación "Proveer ayuda".

| |
|---|
| Caso de prueba:1 |
| No. Historia de usuario: 7 |
| Nombre de Caso de prueba: Proveer ayuda. |
| Descripción: Esta prueba se realiza con el objetivo de comprobar que el sistema provee ayuda. |
| Condiciones de ejecución: El usuario debe estar autenticado. |
| Entradas: El usuario presiona el botón opciones, después presiona el botón ayuda. |
| Resultado esperado: El Sistema debe mostrar la interfaz de ayuda con información sobre la aplicación, vínculos de interés, información del contacto y una leyenda. |
| Evaluación: Prueba exitosa. |

Resultado de las pruebas de aceptación.

En las pruebas de aceptación realizadas a las funcionalidades identificadas por el cliente se encontraron 22 casos de pruebas en total para las 4 iteraciones con un número variable de escenarios para cada caso. En la primera iteración se revisaron 6 casos de prueba detectándose 3 no conformidades a las cuales se les dio solución. En la segunda iteración se revisaron 6 casos de pruebas en los cuales se encontraron 2 no conformidades las que fueron resueltas inmediatamente. En la tercera iteración se verificaron 4 casos de pruebas en los cuales se encontró 1 no conformidad, siendo esta solucionada posteriormente. En la cuarta iteración se revisaron 6 casos de prueba detectándose 1 no conformidad la cual fue solucionada. Posteriormente se realizó una iteración de pruebas adicional para verificar que se resolvieran todas las no conformidades encontradas anteriormente. Esta arrojó 0 no conformidades, lo que indica que el Cliente HMI del SCADA Guardián del ALBA para Dispositivos Móviles cumple con los requisitos establecidos por el cliente y con los objetivos planteados al inicio de su desarrollo.

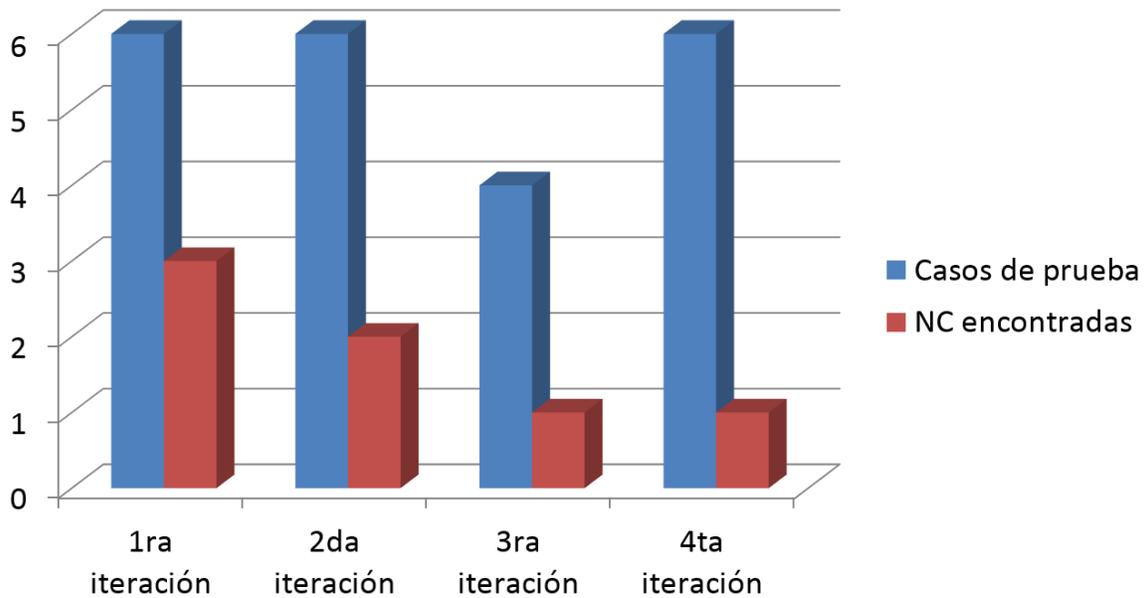


Figura 14: Resultados de las pruebas de aceptación.

3.3 Consideraciones parciales

En las pruebas de aceptación realizadas al software para comprobar la calidad del mismo se evidencia que se cumple los requisitos establecidos por el cliente durante las fases previas a las pruebas.

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 los 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.

Recomendaciones

- Se recomienda incluir al sistema los módulos para visualizar reportes y los despliegues.
- Implementación de un framework para facilitar el desarrollo de interfaces de usuario de software industrial en dispositivos móviles.
- Empaquetar la aplicación para otros sistemas operativos móviles.

Bibliografía

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 lacteos 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, Joaquin Molina y Espejo, Laura Baena.** *Sistemas Operativos en Entornos monousuario y multiusuario.* . Madrid : Vision Net, 2007.
10. **Google.** [www.android.com.](http://developer.android.com/about/index.html) [En línea] Google. [Citado el: 15 de enero de 2014.] <http://developer.android.com/about/index.html>.
11. **Apple.** [www.apple.com.](https://www.apple.com/es/ios/what-is/) [En línea] Apple. [Citado el: 12 de diciembre de 2013.] <https://www.apple.com/es/ios/what-is/>.
12. **Blackberry.** [es.blackberry.com.](http://es.blackberry.com/software/smartphones/blackberry-10-os.html) [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.](http://www.myscada.org/wp/myscada-2/) [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.](https://itunes.apple.com/us/app/prosoft-i-view/id385920646?mt=8) [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.sweetwilliamsl.com.](http://www.sweetwilliamsl.com/scadamobile/features/) [En línea] SweetWilliam. [Citado el: 25 de diciembre de 2013.] <http://www.sweetwilliamsl.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/>.
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.

Glosario de términos

Despliegues: Representación gráfica del proceso productivo supervisado. Una figura simbólica que presenta al usuario un modelo del proceso incluyendo las relaciones entre las variables.

Framework: Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Full-dúplex: Utilizado en las telecomunicaciones para definir a un sistema que es capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea.

GNU/Linux: Es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL y otra serie de licencias libres.

HMI: Una interfaz Hombre - Máquina o HMI ("Human Machine Interface") es el aparato que presenta los datos a un operador (humano) y a través del cual éste supervisa y controla el proceso.

JSON: Acrónimo de JavaScript Object Notation, 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. La simplicidad de JSON ha dado lugar a la generalización de su uso.

Middleware: Es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones que son necesarias en los sistemas distribuidos.

Modbus: Es un protocolo de comunicaciones situado en el nivel 7 del Modelo OSI, basado en la arquitectura maestro/esclavo o cliente/servidor, diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLCs).

Multitouch: La tecnología multitouch consiste en una pantalla táctil o touchpad que reconoce simultáneamente múltiples puntos de contacto, así como el software asociado a esta que permite interpretar dichas interacciones simultáneas.

Ofimática: Equipamiento hardware y software usado para crear, coleccionar, almacenar, manipular y transmitir digitalmente la información necesaria en una oficina para realizar tareas y lograr objetivos básicos.

QWERTY: El teclado QWERTY es el teclado más común. Fue diseñado y patentado por Christopher Sholes en 1868 y vendido a Remington en 1873. Su nombre proviene de las primeras seis letras de su fila superior de teclas.

Simbiosis: El término simbiosis (del griego: σύν syn "con"; y βίωσις biosis "vivir") hace referencia a la relación estrecha y persistente entre organismos de distintas especies.

SmallTalk: Es un lenguaje de programación que permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales. Metafóricamente, se puede considerar que un Smalltalk es un mundo virtual donde viven objetos que se comunican mediante el envío de mensajes.

Trackball: Es un dispositivo apuntador estacionario compuesto por una bola incrustada en un receptáculo que contiene sensores que detectan la rotación de la bola en dos ejes, como si fuera un ratón de computadora boca arriba, pero con la bola sobresaliendo más.

Trackpad: Es un dispositivo táctil de entrada que permite controlar un cursor o facilitar la navegación a través de un menú o de cualquier interfaz gráfica.

Touchscreen: Es una pantalla que mediante un toque directo sobre su superficie permite la entrada de datos y órdenes al dispositivo, y a su vez muestra los resultados introducidos previamente. Este contacto también se puede realizar por medio de un lápiz óptico u otras herramientas similares. Actualmente hay pantallas táctiles que pueden instalarse sobre una pantalla normal.