

Universidad de las Ciencias Informáticas  
“Facultad 5”



**Título: “Visualizador de Reportes del Sistema Guardián del Alba  
en Dispositivos Móviles”**

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

**Autor(es):** Yaimari Nuñez Aldana

**Tutor:** Ing. Leonardo Rafael Fernández Ruiz

**Co-tutor:** Ing. Luanner Kerton Martínez

**Ciudad Habana, Junio 2014**

# Steve Jobs



*"No tenemos la oportunidad de hacer muchas cosas, por lo que cada cosa que hagamos debe ser excelente. Porque esta es nuestra vida".*

### Declaración de Autoría

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 \_\_\_\_\_.

Ing. Leonardo Rafael Fernández

\_\_\_\_\_

Firma del Tutor

Ing. Luanner Kerton Martinez

\_\_\_\_\_

Firma del Co-tutor

Yaimari Nuñez Aldana

\_\_\_\_\_

Firma del Autor

## DATOS DEL CONTACTO

**Tutor:** Ing. Leonardo Rafael Fernández Ruiz, Ingeniero Informático se ha desempeñado como profesor, desarrollador e investigador, en la Línea de Sistemas Embebidos. Actualmente se desempeña como Jefe de Departamento de Desarrollo de Componentes del Centro de Informática Industrial. Ha estado vinculado al área productiva al desarrollo de sistemas de Realidad Virtual, Sistemas Empotrados, Sistemas de Gestión Industrial y SCADAs con 5 años de experiencia.

**Co-tutor:** Ing. Luanner Kerton Martinez, Ingeniero en Ciencias Informáticas que se ha desempeñado como desarrollador e investigador. Ha estado vinculado a áreas productivas de Realidad Virtual y Gestión Industrial. Su principal área de investigación son los sistemas de gestión empresarial y tecnologías web con más de 3 años de experiencia.

## Agradecimientos

- ✓ *A mi madre gracias por siempre estar apoyándome en todo.*
- ✓ *A mis padres Ismaris y Jorge, por darme la dicha de existir y ser la persona más feliz de este mundo al poder contar con ello.*
- ✓ *A mi abuela Dolores (aunque ya no esté con nosotros) por quererme tanto.*
- ✓ *A mis tías y tíos Cristina, Odalis, Idalmis, María, Adonis, Enrique, Lorenzo, Velis por preocuparse tanto por mí y mi carrera.*
- ✓ *A mis primas Yaidalis, Yaima, Ismaralis, Dailen, Aidee, por apoyarme en esto.*
- ✓ *A mi novio por soportarme en los momentos difíciles y de estrés.*
- ✓ *A mi tutor Ing. Leonardo Rafael Fernández Ruiz por apoyarme tanto y dedicar tanto tiempo en mí para que saliera bien.*
- ✓ *A mi Co-tutor Ing. Luanner Kerton que ha sido incondicional conmigo, no sé si le podre agradecer en todo mi vida todo el tiempo que ha empleado en mí.*
- ✓ *A mis compañeros de aula (Amanda, Liena, Gretel, Aylin, Aylen, Anayanci, Elizabeth, Rachel, Manolo, Luis Javier, Carlos, Omar, Frank, Luis Manuel, David, Ernesto, Riveron, Yoan, Rafael, Angel) por soportarnos el uno a los otros. Nunca los olvidaré.*
- ✓ *A la Revolución por darme esta oportunidad.*

## Dedicatoria

- ✓ *Mi esfuerzo durante estos cinco años de la carrera, todos mis resultados como estudiante, mis logros personales y mi título de Ingeniera en Ciencias Informáticas, van dedicados con todo cariño:*
- ✓ *A mi madre, diamante del cofre de mi corazón, por darme la virtud de vivir, crecer y luchar, por apoyarme en los buenos y malos momentos, por estar siempre cuando la necesito.*
- ✓ *A mi padre por ser el faro que ilumina mi vida.*
- ✓ *A mi abuelita que está en el cielo.*
- ✓ *A mi novio por hacerme sentir tan especial.*
- ✓ *Y a toda mi familia, los cuáles significan todo en mi vida.*

## Resumen

La supervisión y control de los procesos industriales se define como una actividad fundamental para el desarrollo de la sociedad actual. Los sistemas de Supervisión, Control y Adquisición de Datos constituyen un elemento de apoyo para la ejecución correcta de procesos en cualquier industria moderna, donde la capacidad de evaluación y monitoreo por parte de operadores, ejecutivos y directivos constituyen un factor estratégico en la administración y toma de decisiones. Por tanto, los reportes a partir de consideraciones de la dirección, y el acceso a éstos desde distintas plataformas y en tiempo real se convierten en tareas primordiales para los desarrolladores de sistemas SCADA. Siguiendo este paradigma, el Centro de Informática Industrial de la Universidad de Ciencias Informáticas desarrolla el Sistema Supervisorio de Procesos Industriales “Guardián del Alba”, SCADA destinado para la empresa Petróleo de Venezuela (PDVSA). En el presente trabajo se expone el desarrollo de un visualizador de reportes para este sistema, que permitirá la supervisión de los procesos controlados por el SCADA desde dispositivos móviles. Se argumentan las bases teóricas que sirven de base para la investigación, su diseño, implementación y validación, según los artefactos propuestos por la metodología *Extreme Programming*.

Palabras claves: dispositivo móvil, reportes, SCADA, visualización,

## Contenido

Introducción .....	1
Capítulo 1. Fundamentación Teórica .....	5
1.1 Reportes .....	5
1.2 Dispositivos móviles .....	6
1.2.1 Características que denotan a los dispositivos móviles .....	6
1.2.2 Limitaciones de los dispositivos móviles .....	7
1.3 Aplicaciones para la generación de reportes .....	8
1.3.1 Dynamicreport.....	8
1.3.2 Crystal Reports .....	8
1.3.3 Generador Dinámico de Reportes Dinámicos (PATDSI GDR) .....	9
1.3.4 FastReport .....	9
1.3.5 Reporting Mobile.....	9
1.3.6 eMD Mobile .....	10
1.3.7 SAP Business One .....	11
1.4 Reportes en el SCADA.....	12
1.4.1 Generador de reportes.....	12
JasperReports .....	12
1.5 Metodología de Desarrollo.....	13
1.5.1 Proceso Unificado de Desarrollo de Software (RUP).....	13
1.5.2 Programación Extrema o XP .....	14
1.6 Lenguaje de modelado .....	16
1.7 Lenguaje de programación y protocolo de comunicación .....	16
1.7.1 HTML5 .....	16
1.7.2 JavaScript.....	17
1.7.3 CCS3.....	17
1.7.4 Protocolo WebSocket .....	17
1.7.5 Phonegap.....	18
1.8 Entorno de desarrollo Netbeans .....	18
1.9 Conclusiones Parciales.....	19

Capítulo 2. Análisis y Diseño del Sistema .....	20
2.1 Consideraciones del negocio .....	20
2.1.1 SCADA “Guardián del Alba” .....	20
2.1.2 HMI Web del SCADA “Guardián del Alba” .....	21
2.2 Generación de reportes en el SCADA “Guardián del Alba” .....	22
2.2.1 Proceso de generación de reportes.....	22
2.3 Objeto de automatización .....	25
2.4 Propuesta de solución .....	26
4.2.1 Selección y análisis de la metodología de desarrollo .....	26
2.5 Exploración y planificación .....	27
2.5.1 La fase de exploración y planificación .....	27
2.5.2 Historias de usuario .....	27
2.5.3 Requisitos no Funcionales .....	32
2.5.4 Estimación de esfuerzo por historias de usuario .....	33
2.5.5 Plan de iteraciones .....	34
2.5.6 Plan de entregas .....	35
2.6 Diseño de la solución propuesta .....	35
2.6.1 Arquitectura .....	36
2.6.2 Interacción cliente servidor .....	38
2.6.4 Mapeo de datos de parametrización .....	42
2.6.5 Tarjetas CRC del lado del servidor.....	43
2.7 Conclusiones parciales.....	44
Capítulo 3 Implementación y Pruebas .....	46
3.1 Implementación.....	46
3.1.1 Tareas de ingeniería .....	46
3.1.2 Vista de despliegue.....	51
3.2 Pruebas.....	52
3.2.1 Casos de prueba .....	53
3.3 Conclusiones parciales.....	59
Conclusiones.....	60

Recomendaciones .....61  
Bibliografía.....62

## Índice de Ilustraciones

Figura 1.1 Vista de Reporting Mobile .....	10
Figura 1.2 Vista de eMD Mobile .....	11
Figura 1.3 Vista de SAP Business One.....	11
Figura 1.4 Fases de RUP.....	14
Figura 2.1 Vista de autenticación de usuarios.....	23
Figura 2.2 Vista de reportes .....	24
Figura 2.3 Configuración de fechas del reporte .....	25
Figura 2.4 Ejemplo de reportes sobre una plantilla .....	25
Figura 2.1 Vista de arquitectura entre el cliente y el servidor .....	36
Figura 2.2 Vista completa del componente de autenticación del cliente .....	37
Figura 2.3 Petición de Autenticación del Cliente al servidor.....	39
Figura 2.4 Respuesta de autenticación del servidor al cliente .....	39
Figura 2.5 Error de autenticación enviado por el servidor.....	40
Figura 2.6. Proceso de solicitud, construcción y envío de vistas.....	41
Figura 2.7 Mecanismo de procesamiento del servidor .....	41
Figura 2.8 Secuencia para la solicitud de reportes.....	42
Tabla 3.1 Vista de despliegue de la solución .....	52
Figura 3.1 Cantidad de errores de Historia de Usuario por iteración .....	53

## Índice de Tablas

Tabla 1.1 Criterios de elección de entre metodologías tradicionales y ágiles .....	27
Tabla 2.1 Historia de Usuario: Autenticar usuario .....	29
Tabla 2.2 Historia de Usuario Visualizar reportes .....	29
Tabla 2.3 Historia de Usuario Visualizar datos de reporte .....	30
Tabla 2.4 Historia de Usuario Filtrar datos de reporte.....	30
Tabla 2.5 Historia de Usuario Salvar datos de reporte a HTML .....	31
Tabla 2.6 Historia de Usuario Salvar datos de reporte a Excel.....	32
Tabla 2.7 Historia de Usuario Salvar datos de reporte a PDF.....	32
Tabla 2. 8 Estimación de esfuerzo a partir de historias de usuario.....	34
Tabla 2.10 Plan de entregas .....	35
Tabla 2.11 Tarjeta CRC Clase Server .....	44
Tabla 2.12 Tarjeta CRC Clase ReportGenerator .....	44
Tabla 3.1 Tareas de ingeniería del sistema .....	47
Tabla 3.2 Tarea de Ingeniería Validar datos de usuario .....	48
Tabla 3.3 Tarea de ingeniería Obtener configuración de reportes .....	48
Tabla 3.4 Tarea de ingeniería Construir vista de reportes .....	49
Tabla 3.5 Tarea de ingeniería Enviar datos de reporte .....	49
Tabla 3.6 Tarea de ingeniería Obtener configuración de reporte.....	50
Tabla 3.7 Tarea de ingeniería Enviar datos de reporte .....	50
Tabla 3.8 Tarea de ingeniería Configurar vista del reporte.....	51
Tabla 3.9 Tarea de ingeniería Construir vista del reporte.....	51
Tabla 3.10 Resumen de errores durante las iteraciones del ciclo de desarrollo. ....	52
Tabla 3.10 Tarjeta de aceptación Autenticar Usuario.....	54
Tabla 3.11 Tarjeta de aceptación Visualizar reportes .....	55
Tabla 3.12 Tarjeta de aceptación Visualizar datos de reportes .....	56
Tabla 3.13 Tarjeta de aceptación Filtrar reporte.....	56
Tabla 3.14 Tarjeta de aceptación Salvar datos como PDF.....	57
Tabla 3.15 Tarjeta de aceptación Salvar datos como Excel.....	58
Tabla 3.16 Tarjeta de aceptación Salvar datos como HTML. ....	58
Tabla 3.17 Tarjeta de aceptación Cerrar sesión de usuario. ....	59

## Introducción

Con el desarrollo de la sociedad, la supervisión y control de procesos industriales marcan una necesidad constante dentro de las Tecnologías de la Información y las Comunicaciones (TIC). La gran demanda de la industria moderna, la humanización de las tareas, la simplificación de actividades de supervisión y recuperación, el ahorro de tiempo y dinero, más las búsqueda de nuevas capacidades tecnológicas, impulsan el desarrollo de herramientas como los SCADA.

Un SCADA (*Supervisory Control and Data Acquisition* por sus siglas en inglés) es un sistema basado en computadores que permite supervisar y controlar a distancia una instalación de cualquier tipo. A diferencia de los Sistemas de Control Distribuido, el lazo de control es generalmente cerrado por el operador. Hoy es común encontrar sistemas de este tipo realizando actividades a todos los niveles dentro de los procesos industriales, aunque su principal misión es la supervisión y control de datos provenientes de los dispositivos de campo comúnmente conocidos como PLC (*Programmable Logic Controller* por sus siglas en inglés).

En el flujo de información en los sistemas SCADA el fenómeno físico lo constituye la variable que deseamos medir. Dependiendo del proceso y la naturaleza del fenómeno pueden ser: presión, temperatura, flujo, potencia, intensidad de corriente, voltaje, densidad, entre otras. Este fenómeno debe traducirse a una variable que sea inteligible para el sistema SCADA, es decir, en una variable digital. Para ello se utilizan los sensores o traductores que llevan la información hasta el sistema para ser convertida en información digital y pueda ser visualizada y manipulada por los operadores.

En la Universidad de las Ciencias Informáticas el Centro de Informática Industrial (CEDIN) tiene como principal producto un sistema SCADA que lleva por nombre Sistema Supervisorio de Procesos Industriales “Guardián del Alba”; actualmente se encuentra en fase de implantación en varias instalaciones de PDVSA en la República Bolivariana de Venezuela. El mismo tiene como propósito automatizar los procesos industriales de aquellas pequeñas, medias y grandes industrias que requieran del control de sus procesos.

Dentro de las actividades propias del SCADA, la visualización de datos recolectados de campo es un área principal a la cual se dedica gran parte del desarrollo, esto debido a la necesidad de presentar a los operadores la información necesaria y con los valores suficientes para que pueda ser interpretada de forma adecuada, tanto para los procesos que se analizan en tiempo real, como aquellos que se obtienen a partir de datos históricos. Este requerimiento tiene particularidades de visualización que son necesarios tener en

cuenta para la supervisión y control de procesos. Por un lado los usuarios requieren observar un diagrama general del proceso de producción lo más realista posible, dando la posibilidad de comenzar a visualizar la información y terminar con una rápida y precisa reacción del operador ante eventos.

En la industria, ejecutivos, supervisores, gerentes y empleados suelen necesitar información en horas específicas para tomar decisiones empresariales críticas, esto requiere que el sistema brinde acceso completo a la información desde distintas plataformas. En este momento, el sistema SCADA “Guardián del Alba” no presenta la facilidad de acceder a la información desde distintas plataformas, siendo necesario que los operadores y supervisores estén en los centros de control para realizar dichas tareas. La falta de acceso trae consigo dificultad en la organización para la planificación, generación de informes, uniformidad de los datos y toma de decisiones, ocasionando:

1. Retraso y esfuerzos duplicados en la generación de los informes: Durante las actividades de análisis de información, los operadores generan reportes constantemente que muchas veces, contiene información duplicada o se realizan de forma innecesaria.
2. Planificación de mantenimientos innecesarios: Es necesaria la planificación de mantenimiento a los centros de control, debido a las cargas constantes de trabajo, afectando la operatividad y disponibilidad del sistema.
3. Desconocimiento por la gerencia de la realidad en la industria: Dada la falta de capacidad de acceso a la información desde distintas plataformas limita su acceso desde la alta gerencia en los momentos deseados, despreciando uno de los principios de supervisión y permitiendo un retraso en la toma de decisiones.
4. Pérdida de tiempo y dinero: Estas incapacidades producen pérdidas a las empresas que obtienen el sistema y limitan los valores de éste durante su percepción en los procesos de venta por parte del CEDIN.

A partir de las nuevas tendencias de las tecnologías y las prestaciones de los sistemas de supervisión, tableros de control y sistemas de mando, la visualización de información de sistemas informáticos industriales en dispositivos móviles (celulares, paginadores, PDA, entre otras.) se convierte en un área en explotación y que resuelve los problemas de acceso, aplicable también a los SCADA. En este sentido, los teléfonos inteligentes han marcado la sociedad haciéndose indispensables como medios de comunicación para las actividades personales y muchas organizaciones han sacado provecho para incluir tareas

laborales. Esta característica los hace candidatos para resolver los problemas de disponibilidad de la información que están presentes en el escenario de las industrias que se supervisan con “Guardián del Alba”, dando la posibilidad de realizar informes sobre una base de datos común para resolver los inconvenientes anteriormente expresados.

La visualización de esta información en dispositivos móviles, tiene gran dimensión debido al cúmulo de datos que presenta el SCADA. En este sentido, los valores de las variables, las gráficas, la información de despliegues, las alarmas o los estados son representativos y relevantes para la toma de decisiones. El análisis total de estos datos es complejo a partir de ambientes diferentes, mucho más cuando se trata de realizar en tiempo real. Una de las soluciones óptimas es diseñar reportes a medida que permita al operador evaluar dinámicamente la información.

A partir de la situación problemática antes expuesta surge el siguiente **problema a resolver**: ¿Cómo contribuir a aumentar el acceso a la información de los reportes del SCADA “Guardián del Alba” para mejorar el proceso de toma de decisiones?

Como **objeto de estudio**: El proceso de visualización de reportes. Se define como **campo de acción**: El proceso de visualización de reportes del Sistema SCADA “Guardián del Alba” para dispositivos móviles.

Definiendo como **objetivo general**: Desarrollar un visualizador de reportes para dispositivos móviles que contribuya a aumentar la disponibilidad de la información del Sistema SCADA “Guardián del Alba”.

### **Posible resultado**

Un visualizador de reportes del SCADA “Guardián del Alba” para dispositivos móviles que permitirá disponer de información actualizada contribuyendo al análisis de información relevante para la toma de decisiones.”

Para alcanzar dichos objetivos se planteó desarrollar las siguientes **tareas de investigación**:

1. Identificación de los elementos teórico-técnico-metodológicos que permitan el desarrollo adecuado de un sistema de visualización de reportes para dispositivos móviles.
2. Caracterización del proceso de generación y visualización de reportes en el SCADA “Guardián del Alba”.
3. Análisis y diseño del visualizador de reportes para dispositivos móviles.
4. Implementación de la solución.

5. Validación del visualizador de reportes mediante el consumo de datos del servidor HMI Web del SCADA “Guardián del Alba”.

Los **métodos científicos** que permitirán el desarrollo de la investigación serán los siguientes:

### **1. Métodos Empíricos:**

Entrevista: Empleando este método se realizan entrevistas al cliente para obtener información que ayude a la satisfacción de sus necesidades, y de este modo conocer los problemas existentes y luego darles solución.

### **2. Métodos Teóricos:**

Histórico-Lógico: Mediante este método se estudia todo lo relacionado con los dispositivos móviles, para así obtener un conocimiento histórico de su desarrollo y comportamiento tanto a nivel internacional como nacional.

Analítico Sintético: Empleando el método analítico sintético es posible analizar los documentos y las teorías referentes a los dispositivos móviles, proporcionando la extracción de los datos más importantes.

La investigación se organiza en tres capítulos siguientes:

**Primer capítulo: Fundamentación Teórica.** Abarca los aspectos teóricos de la investigación, se realiza un estudio del estado del arte detallando aquellas definiciones que tienen relación directa con la solución a desarrollar. Se exploran las técnicas y herramientas seleccionadas para el desarrollo del objetivo propuesto.

**Segundo capítulo: Análisis y Diseño del Sistema.** Se describe el negocio y los requisitos necesarios para alcanzar los objetivos de la investigación; se ofrecerán detalles del diseño del sistema a partir de los artefactos de la metodología propuesta, indicando los planes de entrega y las características de la arquitectura propuesta.

**Tercer capítulo: Implementación y prueba.** Se obtiene un prototipo funcional del sistema y se muestran los resultados de las pruebas realizadas. Se describen los resultados tras la utilización del visualizador en conjunto con el servidor del SCADA.

## Capítulo 1. Fundamentación Teórica

El presente capítulo abarca los aspectos teórico-técnicos de la investigación. Se especifican algunos conceptos asociados al área de aplicación y se hace un estudio del estado del arte para el área nacional e internacional. Se efectúa un análisis de las metodologías de desarrollo, lenguajes de programación y tecnologías que se utilizan para el desarrollo de software.

### 1.1 Reportes

Un reporte es un documento que a partir de los objetivos de quien lo crea, pretende transmitir información. Concretamente, dentro de la estadística y la toma de decisiones. Al concepto de reporte se le añade las siguientes condiciones (1):

- La información debe ser estructurada y organizada: La información obtenida en el reporte debe tener una estructura lógica, con organización coherente que garantice que su receptor puede comprenderla fácilmente.
- Simplificación de las fuentes: Un reporte debe sintetizar la información de las fuentes, sin importar la naturaleza de éstas, añadiendo valor a la información ya sea mediante el resumen, la representación gráfica o la agrupación de elementos de interés.

Los reportes pueden ser de diversos tipos o estar soportados en diversos formatos (documentos impresos, digitales, audiovisual, entre otros.). En el ámbito de la informática, los reportes son informes que se organizan y muestran información a partir de fuentes de datos digitales diversas (textos, bases de datos, etc.). Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios. En esta dimensión, los reportes contienen otras características (2):

- Son generados dinámicamente: Los reportes son extraídos mediante acciones simples sin necesidad de conocer la naturaleza de las fuentes. La selección, agrupación y representación de la información es realizada por el ordenador según las exigencias del usuario. Por ejemplo, un programa que presente tendencias de ventas accede a la base de datos transaccional, obtienen las ventas diarias y construye una gráfica a partir de los valores.

- A diferencia de un formulario, los datos de los reportes no pueden ser modificados: Los reportes representan información de salida que no puede ser reajustada para ingresarlas nuevamente a las fuentes. Esta información debe ser modificada en otras partes del sistema para que sus cambios se vean reflejados cuando se genera el reporte nuevamente.
- Los reportes son específicos por áreas de interés dentro del sistema: El reporte contiene la capacidad de crear etiquetas y responder a éstas. Por ejemplo, una etiqueta de compras permite reflejar el proceso de compras de una empresa a través de reportes gráficos o resúmenes en tablas.

La representación del reporte puede ser a partir de formatos diversos y para distintas plataformas, desde ordenadores convencionales hasta dispositivos móviles.

### 1.2 Dispositivos móviles

Un dispositivo móvil es un aparato electrónico que se define por: ser de pequeño tamaño, capacidad especial de procesamiento, conexión a una red de datos, memoria limitada, versatilidad y fundamentalmente, es de uso personal. Un concepto importante es la movilidad, ya que define su capacidad de movimiento y que pueden ser empleados funcionalmente durante su transporte. La capacidad de móvil e inalámbrico suele confundirse, pues aunque en su mayoría presentan conexión inalámbrica, existen algunos dispositivos móviles alámbricos, como el caso de algunos PDA (*Personal Digital Assistant* por sus siglas en inglés) (3).

#### 1.2.1 Características que denotan a los dispositivos móviles

Las características fundamentales son (3):

- 1 Funcionalidad limitada.
- 2 No necesariamente extensible y actualizable.
- 3 Vida útil limitada a pocos años que propicia un cambio constante.
- 4 Tecnología de costos más bajo que los dispositivos no móviles.
- 5 Menos complicado en su manejo.
- 6 Fácil de aprender su operación.
- 7 No se requieren usuarios expertos.

Existe gran diversidad de dispositivos móviles, los más destacados son:

- Paginadores.
- Comunicadores de bolsillo.
- Teléfonos con pantalla para Internet.
- Sistemas de navegación de automóviles.
- Sistemas de entretenimiento.
- Sistemas de televisión e Internet.
- Teléfonos móviles.
- Tabletas.
- Organizadores y asistentes personales digitales.

### 1.2.2 Limitaciones de los dispositivos móviles

Los arquitectos de soluciones para dispositivos móviles deben considerar una cantidad de factores limitantes, muchos más que al generar aplicaciones para una computadora personal. Algunos desafíos que se enfrentan al construir arquitecturas en el entorno móvil son (4):

- 1 Los dispositivos tienen recursos limitados. A medida que los dispositivos y sus componentes pasan a ser más pequeños y livianos, la energía disponible para respaldar aplicaciones ricas disminuye. También, la capacidad de la memoria del dispositivo es limitada, aunque esto se mitiga mediante mejoras en la tecnología de almacenamiento.
- 2 Los dispositivos no están estandarizados. Los arquitectos de soluciones móviles deberán optimizar el software y el hardware para obtener experiencias más completas. La falta de estandarización del hardware obliga a los desarrolladores a suponer el grupo de recursos que estarán disponibles para la aplicación.
- 3 Los dispositivos deben soportar escenarios sin conexión y a veces, conexiones lentas. Necesitan ser más que pantallas de cliente liviano; deben ser plataformas de aplicación por sí mismos. Un requisito clave para esta capacidad es el almacenamiento y procesamiento local, con sincronización con ordenadores o servicios de Internet.
- 4 La conectividad no está estandarizada. En función de las capacidades de los dispositivos y del plan de servicios que tiene con su operador de red, un usuario puede acceder a información y servicios en

Internet móvil a través de la voz, mensajería (mensajes cortos, correo electrónico) o a través de protocolos de Internet. En los mercados emergentes es probable que los servicios deban ofrecerse sobre SMS (siglas del inglés *Short Message Service* o Servicio de Mensajes Cortos en español) o voz ya que es más usual que los usuarios posean teléfonos masivos con capacidades muy limitadas. Para experiencias ricas de multimedia se necesita un protocolo de Internet rápido.

No obstante a estas divergencias planteadas principalmente por el modelo tecnológico ofrecido por los fabricantes, existen estándares de comunicación que ofrecen un mecanismo común para toda la gama de dispositivos del mercado. En este caso se pueden citar los protocolos de visualización de datos (imágenes, videos, gráficos, etc.) y los protocolos de conectividad, como el *IEEE 802.11 (Wi-Fi)* e *IEEE 802.15.1 (Bluetooth)*.

### **1.3 Aplicaciones para la generación de reportes**

#### **1.3.1 Dynamicreport**

Dynamicreport es un proyecto comunitario rectorado por la línea JEE del Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE), su principal objetivo es crear un reporteador dinámico que pueda abstraer al programador de la fuente de datos. Está basado en el funcionamiento de JasperReport y DynamicJasper. El proyecto es web, y puede ser utilizado de forma independiente o como un componente del framework<sup>1</sup> Dalas. Soporta como acceso a datos JDBC, Hibernate y XML. Permite generar reportes en archivos pdf, csv, xls entre otros (5).

#### **1.3.2 Crystal Reports**

Crystal Reports es la solución de elaboración de informes más usada en el mundo, con más de 8 millones de copias vendidas (6). Es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos. Varias aplicaciones, como Microsoft Visual Studio, incluyen una versión OEM (fabricante de equipamiento original) de Crystal Reports como una herramienta de propósito para generar los informes y reportes. Crystal Reports se convirtió en el generador de reportes para informes estándar cuando Microsoft lo liberó con Visual Basic (7).

---

<sup>1</sup> Framework: Marco de trabajo que simplifica el desarrollo de aplicaciones (41).

### **1.3.3 Generador Dinámico de Reportes Dinámicos (PATDSI GDR)**

Es una aplicación web que tiene como objetivo generar reportes de forma rápida, interactiva y con una amplia gama de alternativas para los usuarios. La extensión en su uso puede estandarizar la generación de reportes en diferentes aplicaciones independientemente del Sistema Gestor de Base de Datos que utilicen ya sea MySQL, Oracle o PostgreSQL (8).

El Generador Dinámico de Reportes es una herramienta multiplataforma, que da cobertura al ciclo de vida completo de un reporte: diseño, visualización, almacenamiento, eliminación, etc. Está formado por 4 módulos: Diseñador de Modelos, Diseñador de Reportes, Diseñador de Consultas y Visor de Reportes. Es un proyecto desarrollado en el centro DATEC, varios son los clientes que actualmente hacen uso del producto, entre ellos los centros CEIGE (Sistema de Gestión de Incidencias) y CALISOFT (Centro de Calidad de Software), entre otros, al tiempo que se encuentra en 11 de las soluciones convenidas como parte de la X Comisión Mixta Cuba-Venezuela (9).

### **1.3.4 FastReport**

Tecnologías móviles modernas proporcionan el acceso mundial a la información. FastReport.Net trae informes en las tabletas o teléfonos del usuario y le ayuda a comunicarse con los cuadros de diálogo. FastReport.Net soporta la visualización en navegadores móviles con interfaz de usuario táctil orientada, pero se debe tener en cuenta las características de los sistemas móviles en el desarrollo de informes. En particular, es necesario utilizar una fuente que esté presente en todas las plataformas móviles (10).

### **1.3.5 Reporting Mobile**

Esta aplicación incluye un subconjunto de aplicaciones completas con formatos de informes pre-definidos. Los informes incluyen: Operación Resumen, Resumen Complicación, Resumen Co-morbilidad y más (11).



Figura 1.1 Vista de Reporting Mobile

### 1.3.6 eMD Mobile

Permite acceder a los datos utilizando el *smartphone* o el iPad. Dispone de visualización de datos en tiempo real desde cualquier lugar diseñado para ambientes médicos. Este nuevo e innovador producto ofrece la misma funcionalidad que EMD central en un formato completamente re-escrito en los teléfonos inteligentes e iPads por igual. Se puede acceder a los datos demográficos de los pacientes, los detalles del procedimiento, las complicaciones, las visitas de seguimiento, así como ejecutar informes. El cirujano puede introducir registros de operación en el teléfono inteligente o plataformas iPad desde cualquier lugar, siempre y cuando tengan una conexión de red celular o *Wi-Fi* (12).



Figura 1.2 Vista de eMD Mobile

### 1.3.7 SAP Business One

SAP Business One es una solución económica que administra todo el negocio de una empresa a través de finanzas, ventas, clientes y operaciones, todo en un solo sistema. Su aplicación móvil para iPhone le permite al usuario mantenerse en contacto con sus datos de negocio y de los clientes, aún fuera de la oficina. Le permite además conectar inmediatamente con la aplicación SAP Business One a través de iPhone para comprobar el inventario y clientes, informes de acceso, recibirás un email, y las aprobaciones de procesos (13). Entre las mayores ventajas se encuentran que:

- Recibe alertas de eventos específicos, tales como desviaciones de descuentos aprobados, precios o límites de crédito.
- Utiliza la interfaz intuitiva para clasificar las alertas por fecha, tipo o prioridad.
- Profundiza en los datos pertinentes antes de tomar una decisión.
- Solicita información adicional o formula observaciones en el proceso de aprobación.



Figura 1.3 Vista de SAP Business One.

## 1.4 Reportes en el SCADA

La generación de reportes en los sistemas SCADA siempre ha sido una tarea de importancia. Generalmente se hace necesario contar con un mecanismo capaz de generar reportes referentes a la adquisición y registro de datos, generación de alarmas, entre otros, siendo esto una parte fundamental para la toma de decisiones en las distintas industrias.

Existen en el mundo una gran diversidad de proyectos inmersos en la creación de generadores de reportes, aunque es cierto que en su mayoría están enfocados en la generación de reportes para todos los sectores y no específicamente en ambientes industriales. Los generadores de reportes aplicados específicamente a los sistemas SCADA deben cumplir ciertos parámetros, entre ellos la seguridad, ya que cada usuario debe ver solamente lo que se le está permitido.

### 1.4.1 Generador de reportes

#### JasperReports

Es un motor de generación de reportes de código abierto. Está escrito completamente en Java y es capaz de utilizar los datos procedentes de cualquier tipo de fuente de datos y producir documentos que se pueden visualizar, imprimir o exportar en una variedad de formatos de documentos incluyendo HTML, PDF, Excel, OpenOffice y Word. Esta biblioteca está liberada bajo la Licencia Pública General Reducida (LGPL) (14).

JasperReports se usa comúnmente con iReport, un editor gráfico de código abierto para la edición de reportes. Este editor se encuentra bajo la Licencia Pública General (GPL), por lo que es Software libre. El diseñador iReport cuenta con excelente apariencia visual y permite insertar numerosos componentes dados las facilidades de Java. Tanto JasperReports como iReport pueden usarse en sistemas operativos como Linux y Microsoft Windows.

El sistema SCADA “Guardián del Alba” utiliza este motor para la generación de reportes, a partir de un componente de diseño que permite a los operadores detallar el reporte, y otro que asegura la obtención de los datos de reportes según las especificaciones del diseño.

## 1.5 Metodología de Desarrollo

La mayoría de metodologías de desarrollo de software se clasifican en dos grupos según los manifiestos de sus autores, las metodologías pesadas, que dentro de ellas se encuentra Proceso Unificado de Desarrollo de Software (RUP) y las metodologías ligeras/ágiles dentro de la cual se encuentra Programación Extrema o XP. Estas dos metodologías son las más utilizadas ya que poseen características que hacen más fácil el proceso de desarrollo de software (15).

### 1.5.1 Proceso Unificado de Desarrollo de Software (RUP)

El Proceso Unificado de Desarrollo es un proceso de ingeniería planteado por Kruchten en 1996 cuyo objetivo es *producir software de alta calidad, lo que significa desarrollar un software que cumpla todos los requerimientos planteados por el usuario dentro de una planificación y presupuesto establecido, cubriendo todo el ciclo de vida del producto* (16). Es una metodología lo suficientemente robusta como para desarrollar grandes proyectos y lo suficientemente flexible como para adaptarse a pequeños proyectos (17).

Al ser una metodología naciente de la crisis del software de los años 90, toma en cuenta las mejores prácticas de desarrollo, como la de una arquitectura basada en componentes, el manejo de requerimientos, el proceso iterativo, la modelación visual del software y el control de cambios (15). Como características fundamentales se definen (17) :

**Dirigido por casos de uso:** El proceso de desarrollo se dirige a partir de los casos de uso identificados durante la definición del negocio. Al dirigirse por caso de uso RUP permite un desarrollo organizado, separando las funcionalidades en paquetes y contribuye a una documentación del proyecto.

**Iterativo e incremental:** El software se construye por etapas, iterando según la relevancia de los requisitos, teniéndose un desarrollo incremental con entregas probadas al cliente.

**Centrado en arquitectura:** La línea base de la arquitectura se define al inicio de proyecto y sobre ella se desarrolla el sistema. Se obtiene un producto robusto con una arquitectura bien documentada.

RUP tiene 4 fases definidas, en las que intervienen 7 flujos de trabajo o disciplina que agrupan en forma lógica las actividades a realizar. De estos flujos 4 son principales y 3 de apoyo.

Las fases son (18):

- **Comienzo o inicio:** Se describe el negocio y se delimita el proyecto, describiendo el alcance del mismo con la definición de los casos de uso del negocio.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene un prototipo que responde a los requerimientos identificados.
- **Construcción:** Se obtiene un producto listo para la utilización que está documentado y tiene manual de usuario. Se obtienen tantas versiones de entrega como determinan las pruebas.
- **Transición:** Con la liberación, ya el producto está listo para su instalación en condiciones reales. Puede implicar la reparación de errores como parte del proceso de soporte.

Una descripción gráfica del proceso de desarrollo utilizando RUP puede documentar el esfuerzo empleado por el equipo en cada fase del proyecto (19):

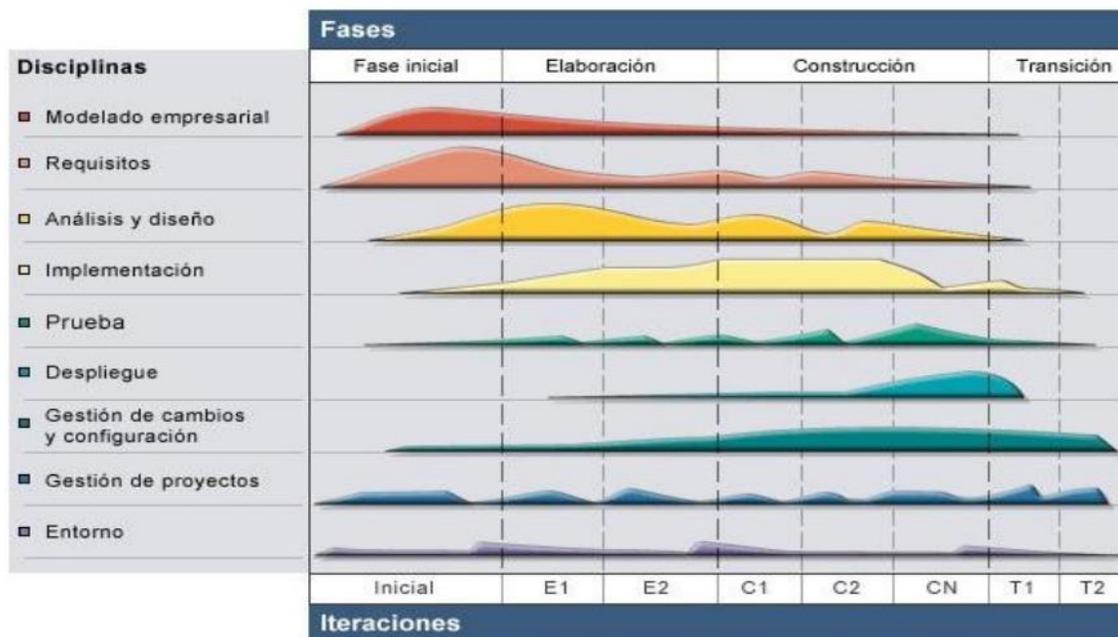


Figura 1.4 Fases de RUP

### 1.5.2 Programación Extrema o XP

*Extreme Programming (XP)* se basa en la necesidad imperante de dar respuesta a los cambios que ocurren durante el proceso de desarrollo de un software, por lo que simplifica procesos de metodologías pesadas y promueve el producto por encima de la documentación. Se puede considerar la programación extrema

como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software (20). Sus principios son:

- 1 **Comunicación:** Potencia el desarrollo rápido y el intercambio con el cliente, se enfoca en relacionar todo el equipo de trabajo y al cliente de forma sencilla y directa (19).
- 2 **Simplicidad:** Enfocado en diseños sencillos del código, en su mayoría autogenerado, con una documentación indispensable y con terminologías fáciles y concretas.
- 3 **Retroalimentación:** Propicia el protagonismo del cliente en el proyecto, la realización de ciclos cortos de desarrollo y la realización de pruebas unitarias apoyados en herramientas que ayuden al flujo correcto del código fuente.
- 4 **Coraje:** El equipo de desarrollo está preparado para cambios o decisiones complejas que afecten al proyecto, y mantienen la mentalidad positiva durante todo el ciclo apoyados en su adaptabilidad y previsibilidad.
- 5 **Respeto:** Se estima toda la magnitud del trabajo de todos los miembros del equipo.

XP propone 4 fases de desarrollo que son:

- **Planificación y estimación:** Se analiza el negocio, se identifican los requisitos y se establecen los planes de entrega.
- **Análisis y diseño:** Se diseña el software, se establece la arquitectura y se modelan las clases del sistema.
- **Implementación:** Se desarrolla el software a partir de las clases modeladas.
- **Pruebas:** Se prueba la solución y se verifica el correcto funcionamiento del sistema.

La metodología se basa en:

- 1 **Pruebas unitarias:** Son pruebas que se realizan al inicio del proceso, identificando las posibles fallas que puedan ocurrir.

- 2 **Reutilización:** La reutilización de código está presente durante todo el proceso de desarrollo, reutilizando los componentes comunes de iteraciones anteriores.
- 3 **Programación en pares:** Una particularidad de la metodología es la programación por pares, fundamentada en la teoría de que dos programadores contribuyen de forma ordenada al perfeccionamiento del código y a la identificación de errores.

El estudio de ambas metodologías permitirá seleccionar la más adecuada según las características del proyecto.

### 1.6 Lenguaje de modelado

El Lenguaje Unificado de Modelado (*UML*) constituye un lenguaje que permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Se compone de diversos elementos gráficos que se combinan para conformar diagramas. Este lenguaje dispone de reglas para combinar tales elementos y permite la modelación de sistemas con tecnología orientada a objetos. Los diagramas son entes importantes de UML, cuya finalidad es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Un modelo UML describe lo que supuestamente hará un sistema, no cómo implementarlo. El modelo gráfico de *UML* tiene un vocabulario en el que se identifican: elementos, relaciones y diagramas (15).

Para modelar, UML clasifica sus diagramas en tres tipos: diagramas de estructura estática que comprenden los diagramas de clases, objetos y casos de uso; diagramas de comportamiento entre los que se encuentran los diagramas de actividades, estados e interacción y por último los diagramas de implementación donde están los diagramas de componentes y despliegues (21).

### 1.7 Lenguaje de programación y protocolo de comunicación

#### 1.7.1 HTML5

*Hyper Text Markup Language* (Lenguaje de Marcado de Hipertexto) más conocido por las siglas HTML. Es un lenguaje de marca diseñado para estructurar textos y presentarlos en forma de hipertexto, es el formato estándar de las páginas web gracias a Internet y a los navegadores, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para web.

Puede ser creado y editado con cualquier editor de textos básico o cualquier otro editor que admita texto sin formato. Utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador. Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (<>), y algunas tienen atributos que pueden tomar algún valor (22).

### 1.7.2 JavaScript

Lenguaje de programación del lado del cliente (en el navegador) creado por BrendanEich en la empresa *Netscape Communications*. Usado en las páginas web para agregar mayor funcionalidad, interacción o animaciones entre otros usos posibles. JavaScript (JS) es muy fácil de aprender para quien ya conoce lenguajes similares como el C++ o Java, pero, dada su simplicidad sintáctica y su manejabilidad, no es tampoco difícil para quien se acerca por primera vez a este lenguaje. Sin embargo, algunos autores lo consideran también una desventaja porque la simplicidad se basa en una disponibilidad de objetos limitada, por lo que algunos procedimientos, aparentemente muy sencillos, requieren rutinas bastante complejas. Es un lenguaje interpretado por lo que no requiere compilación y es además orientado a eventos. No está basado en clases, pero incluye los elementos necesarios para poder crearlas, realizar herencia y programar orientado a objetos (22).

### 1.7.3 CCS3

CSS es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la forma de separar los contenidos y su presentación y es indispensable para la creación de páginas web complejas dándole un nivel de organización que permite el desarrollo de éstas. CSS3 es el último estándar de CSS. CSS3 es compatible hacia atrás completamente con versiones anteriores de CSS (23).

### 1.7.4 Protocolo WebSocket

La especificación de HTML5 WebSockets, define la API (*Application Programmer Interface* por sus siglas en inglés) de sockets que permite a las páginas web usar el protocolo *Secure Socket Web*, para comunicación bidireccional con un host remoto. Se presenta la interfaz WebSocket y define un canal *full-duplex*, que opera a través de una sola toma en la Web. HTML5 Web Sockets pone a disposición del

programador una conexión de socket, a través de Internet, con una sobrecarga mínima proporcionando una enorme reducción del tráfico innecesario en la red y disminuyendo la latencia. Estas soluciones se utilizan a menudo para enviar datos en tiempo real a los clientes, o incluso simular una conexión bidireccional, para mantener dos conexiones *HTTP*. Para conectarse desde cliente web a un servidor utilizando HTML5 Web Sockets, se crea una instancia nueva *WebSocket* y se le pasa la dirección *URL* del servidor. La especificación define un *ws://* y un *wss://*, esquemas para indicar las conexiones *WebSocket* y las conexiones *WebSocket* seguras, respectivamente. Una conexión *WebSocket* se establece mediante la actualización del protocolo *HTTP* al protocolo *Secure Socket Web*, durante la conexión inicial entre el cliente y el servidor, sobre la misma conexión subyacente *TCP/IP* (24).

La elección de los lenguajes y el protocolo de comunicación, se basa en la compatibilidad necesaria con los artefactos y componentes de la versión web de la HMI (*Human Machine Interface* por sus siglas en inglés) del SCADA “Guardián del Alba”, para garantizar la integración en las tecnologías y herramientas que propicien una evolución, comprensión y uniformidad del sistema.

### 1.7.5 Phonegap

Phonegap es una plataforma de código abierto que permite la generación de aplicaciones móviles a partir de tecnologías web que pueden integrar HTML, JavaScript y CSS. Esta plataforma garantiza una aplicación compatible con diversas familias de dispositivos móviles, utilizando interfaces programables que son interpretadas por los dispositivos, viéndolas como aplicaciones nativas sin importar la lógica con que fueron construidas. Entre sus mayores ventajas se encuentra que permite la manipulación de los elementos del aparato móvil (cámara, sensores, etc.) y que la aplicación resultante se genera según la lógica del formato de destino. Además pueden compilarse aplicaciones para distintas plataformas móviles, como BlackBerry, Android y iOS, además de utilizar sus extensiones para crear configuraciones JAVA de alta compatibilidad con los dispositivos del mercado (22).

### 1.8 Entorno de desarrollo Netbeans

NetBeans es una plataforma para el desarrollo de aplicaciones de escritorio usando el lenguaje Java y un entorno de desarrollo integrado (*IDE*) para desarrollar bajo esta plataforma, admite otros lenguajes de

programación como C y C++ mediante los cuales se pueden crear aplicaciones gráficas. La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de Java escritas para interactuar con la API de NetBeans y un archivo especial (manifest) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software (25).

### **Ventajas**

- La plataforma Netbeans puede ser usada para desarrollar cualquier tipo de aplicación.
- Reutilización de Módulos.
- Permite el uso de la herramienta *Update Center Module*.
- Instalación y actualización simple.
- Incluye Templates y Wizards.
- Posee soporte para PHP.

### **Desventajas**

- Poca existencia de pluguins para esta plataforma.
- Hace falta documentación del Rich Client Plataform (RCP).
- No posee un editor de código HTML.

## **1.9 Conclusiones Parciales**

En el presente capítulo, se realizó un estudio de los principales conceptos de reportes y los dispositivos móviles que permitieron establecer las bases teóricas y fundamentos teóricos de la investigación. Además, se detallaron las tecnologías y herramientas necesarias para la solución, estableciendo una base técnica que guiará todo el proceso de desarrollo.

## Capítulo 2. Análisis y Diseño del Sistema

En el presente capítulo se realiza una descripción del negocio donde se enmarca la investigación, así como el objeto de automatización y la propuesta de solución. Se definen los artefactos de las fases de exploración y planeación, análisis y diseño según la metodología XP y se detalla los elementos de la arquitectura propuesta.

### 2.1 Consideraciones del negocio

#### 2.1.1 SCADA “Guardián del Alba”

El término SCADA es utilizado para describir los sistemas encargados de la supervisión y control de procesos industriales, a partir de técnicas de automatización que incluyen, en la mayoría de los casos, un componente programable que actúa sobre la parte operativa de una industria (26). Otros autores incluyen en esta definición, su capacidad para operar a distancia, su capacidad de retroalimentación en tiempo real y la habilidad de controlar automáticamente el proceso en donde actúa (27). Los sistemas SCADA son un principio novedoso para la supervisión de procesos complejos, y se integran de forma coherente con otras técnicas de la ingeniería, cómo la electrónica y la robótica.

Estas características permiten ver al SCADA como una aplicación que provee una interfaz única entre un proceso de campo y sus operadores, a través de un dispositivo tecnológico, un ordenador en este caso, borrando fronteras de lenguaje y distancia.

SCADA “Guardián del Alba” es un Sistema de Supervisión, Control y Adquisición de Datos, desarrollado por el Centro de Informática Industrial de la Universidad de las Ciencias Informáticas para la Empresa Socialista de Capital Mixto Guardián del Alba SA. Se tiene previsto que dicho sistema sea el responsable de la supervisión y control del proceso industrial de la Empresa Petróleos de Venezuela S.A. (PDVSA). Desarrollado a partir del 2006, está enriquecido con los elementos fundamentales de un sistema SCADA de alto nivel, y se ha adaptado a otras necesidades de automatización, como el caso del SCADA para el sector Eléctrico o el SCADA SAINUX (Sistema SCADA para las industrias cubanas).

Al igual que un sistema SCADA tradicional, contiene como módulos una base de datos histórica, un capa *middleware* que comunica los procesos, un componente de seguridad, un módulo de procesamiento, un ambiente de configuración y una interfaz de presentación dirigida al usuario.

Este módulo de representación, más conocido como HMI pertenece al contexto de los Sistemas de Computación y Electrónica, definido como una capa intermedia que independiza y permite la comunicación entre ambas partes (28). En los sistemas SCADA tiene otra función específica: Mostrar a los operadores información detallada acerca de los procesos industriales donde incide el sistema y permitirle, a través de rutinas predefinidas, tomar el control de ellos.

En estos sistemas, a partir de la necesidad de estandarizar la manera de monitorear y controlar un funcionamiento de dispositivos y variables, se crea una marcada utilización de componentes HMI, bajo normas y principios de operatividad, facilidad y unicidad. Las Unidades Remotas y los PLC, no presentan de forma clara la información de relevancia, de forma única y funcional, por lo cual dejan al HMI la responsabilidad de representación y control general de los sistemas, convirtiéndolo en un componente de vital importancia en la gestión de procesos industriales.

En la actualidad los sistemas SCADA poseen varios HMI. Estas pueden ser orientadas al control en los nodos centrales, a través de sistemas de escritorio con GUI<sup>2</sup> altamente funcionales que acaparan todas las responsabilidades del sistema; o basadas en web, que distribuyen dichas operaciones y responsabilidades a través de la red para su control a distancia. Por tanto, el SCADA “Guardián del Alba” tiene varias funcionalidades orientadas a la web.

### **2.1.2 HMI Web del SCADA “Guardián del Alba”**

Un HMI Web es un conjunto cliente-servidor de tipo web, que se comunican utilizando un protocolo de transmisión de datos. Esto permite, en el caso de un SCADA, que pueda brindar sus funciones convencionales de supervisión y control a través de servicios web, aprovechando las ventajas de disponibilidad que esto conlleva.

Siguiendo el principio de que un cliente es una aplicación u ordenador que consume un servicio remoto de otro computador, a través de una red operativa, o un mecanismo de comunicación a distancia, el cliente web consume servicios a través de la red, generalmente sobre un navegador estándar, aunque también pueden hacerlo utilizando otras tecnologías. Estos clientes les permiten a los usuarios interactuar con el proceso, presentan componentes ligeros que le ofrecen facilidad de instalación, y representan nodos

---

<sup>2</sup> GUI: siglas del inglés *Graphical User Interface*.

pequeños que aun así, pueden asumir la responsabilidad de ofrecer a los usuarios una interfaz completamente funcional para el SCADA.

Por su parte, el servidor es un programa que espera peticiones de servicio desde el cliente, las recibe, las procesa y responde al cliente la respuesta solicitada. Es una capa que no interactúa con el operador. A éste servidor llega la información desde los servidores de configuración, se comporta como un HMI convencional, solo que orientada a brindar servicios web.

El agente XStormClient es el encargado de visualizar la información del SCADA “Guardián del Alba” en la web, permitiendo a directivos y personal autorizado de la empresa, en este caso PDVSA, acceder a dicha información desde dispositivos portátiles y estaciones de trabajo, que no tengan instalado dicho sistema y utilizando cualquier sistema operativo (29).

Este componente permite a los usuarios supervisar todos los procesos del SCADA, aprovechando las ventajas de la web, cruzando horizontalmente al sistema y relacionándose con la información brindada por todos los módulos, que es obtenida a través del XStormServer, servidor web para el visualizador Web del SCADA “Guardián del Alba”.

Sin embargo, este conjunto se orienta principalmente a la visualización de despliegues, puntos, alarmas, eventos y gráficas como son las de tendencia, siendo impreciso para la representación de otro tipo de información, como el caso de los reportes.

## **2.2 Generación de reportes en el SCADA “Guardián del Alba”**

Un reporte es un conjunto de datos organizado que brinda información a partir de un repositorio operativo de un proceso de negocio (30). En un sistema SCADA, los reportes adquieren un alto nivel de relevancia gracias a su capacidad de contener resúmenes de datos históricos del funcionamiento del proceso industrial que puede ser utilizado para el análisis del sistema y la toma de decisiones.

### **2.2.1 Proceso de generación de reportes**

Para un operador obtener un reporte, debe seguir ciertos pasos que permiten la obtención de datos particulares de forma dinámica a partir del archivo histórico del SCADA.

### 1. Diseño de reportes

El diseño es el proceso donde el operador escoge aquellas variables que desea reportar. El ambiente de diseño se realiza dentro de la interfaz de escritorio de HMI, e incluye herramientas para la selección, definición y configuración de los componentes deseados por el operador o supervisor. Una vez configurado un reporte, se almacena en forma de consulta y obtiene de la base de datos los datos solicitados.

### 2. Autenticación de usuarios

Dentro del módulo de seguridad del SCADA “Guardián del Alba” se tienen definidos los permisos del usuario dentro de la aplicación. Cuando un usuario necesita visualizar los reportes, es preciso que se autentique desde la interfaz para que el sistema compruebe si tiene derecho o no a la visualización de reportes.

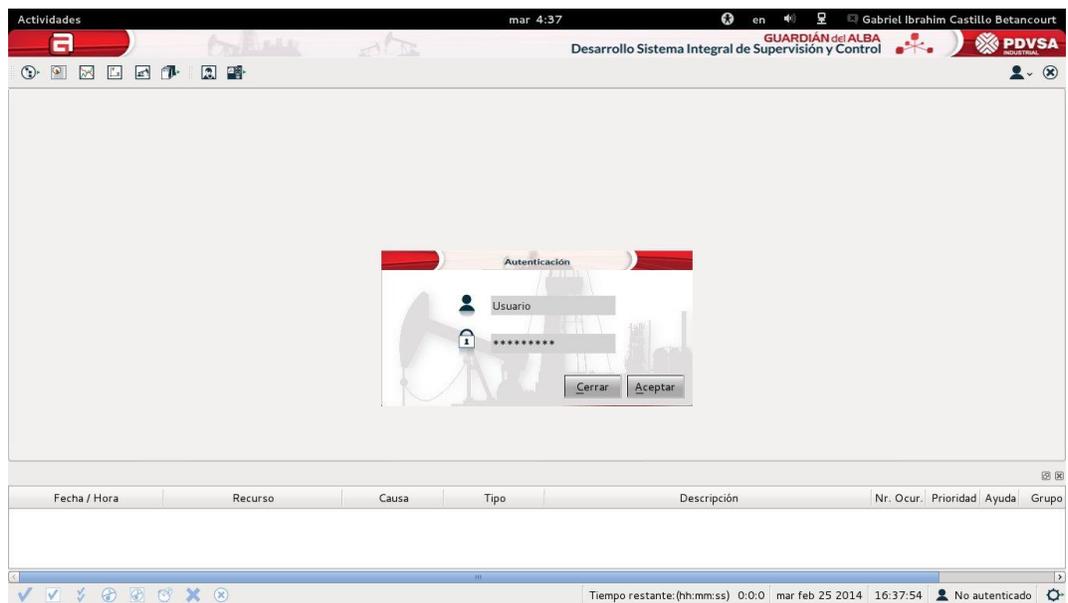


Figura 2.1 Vista de autenticación de usuarios.

### 3. Vista de reportes

Una vez autenticado, dentro de la barra de herramientas el usuario accede a la vista de reportes. En esta vista se muestra un listado de todos los reportes configurados para el sistema. La vista

contiene un paginado que permite ejecutar filtros y definir la cantidad de reportes por página. El usuario puede seleccionar entonces que reporte desea visualizar.

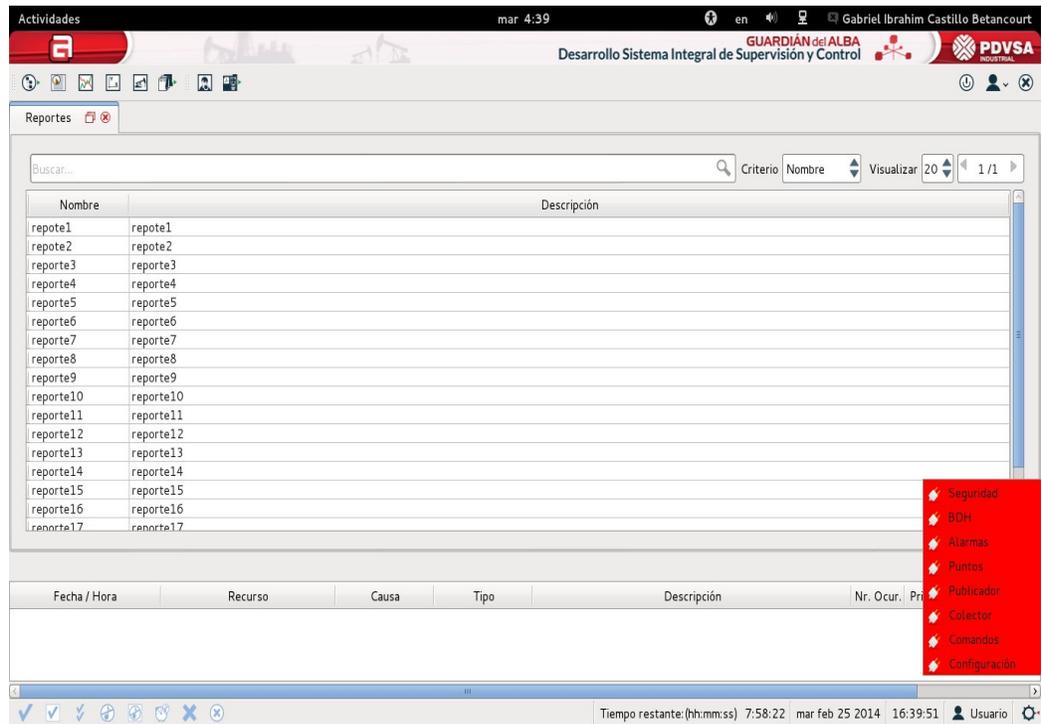


Figura 2.2 Vista de reportes

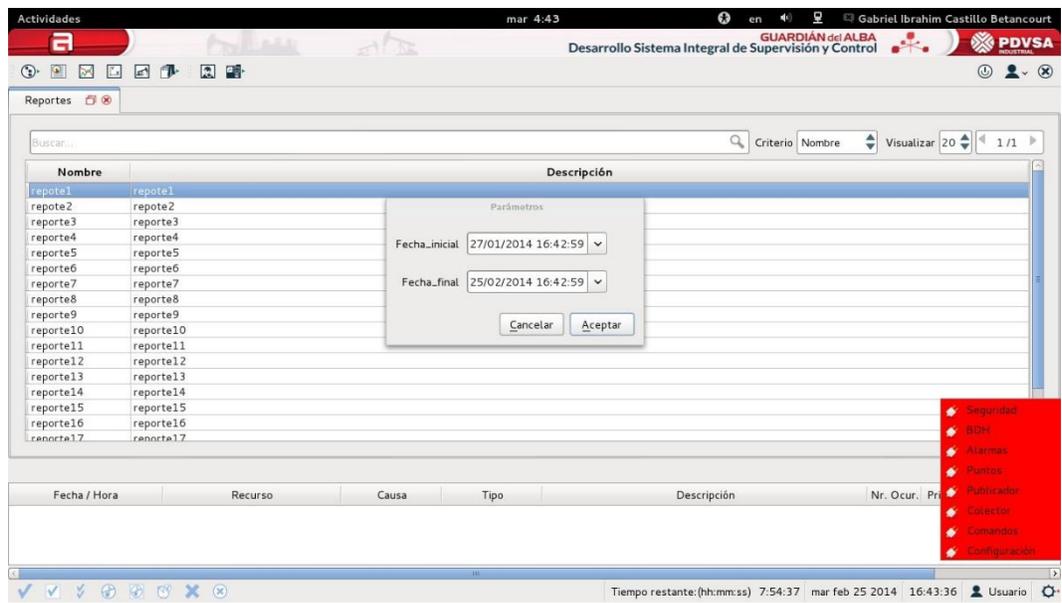


Figura 2.3 Configuración de fechas del reporte

#### 4. Visualización del reporte

Una vez seleccionado el reporte deseado, el sistema muestra los detalles del reporte de acuerdo a su diseño previo. Los datos son mostrados en forma de lista, sobre una plantilla predefinida que puede ser guardada en diferentes formatos según la configuración definida en tiempo de edición.

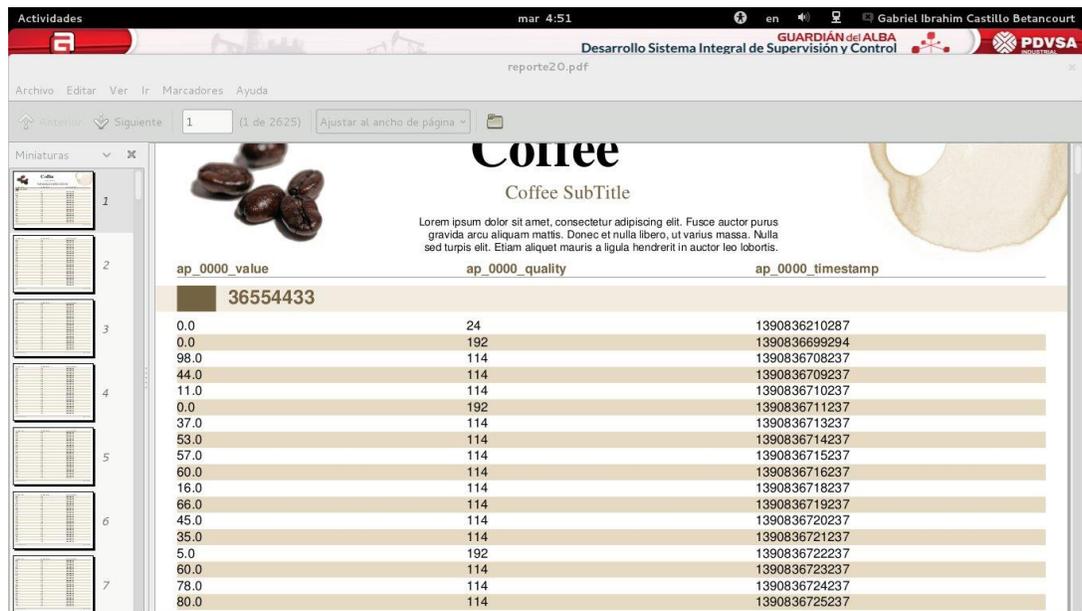


Figura 2.4 Ejemplo de reportes sobre una plantilla

### 2.3 Objeto de automatización

A partir de las consideraciones del epígrafe anterior, se denota la importancia de los reportes en las tareas de supervisión y control de los procesos industriales en los que participa el SCADA “Guardián del Alba”. La visualización adecuada de reportes debe ser comprendida como una actividad imprescindible dentro de la interfaz hombre-máquina que permitan en mayor medida, maximizar el valor de la información recolectada en las bases de datos históricos.

Actualmente, el cliente web del SCADA “Guardián del Alba” no tiene un componente que le permita visualizar los reportes diseñados por los operadores”. Una de las deficiencias que esto aporta, es la

incapacidad de los dispositivos móviles de visualizar adecuadamente los reportes generados por el sistema, limitando las capacidades de disponibilidad necesaria para la herramienta, convirtiéndose en el objeto de automatización de la presente investigación

## 2.4 Propuesta de solución

Se propone el desarrollo de un paquete de clases para los dispositivos móviles que permita la obtención de reportes mediante la conexión al servidor de servicios web del SCADA “Guardián del Alba”. Este componente se ejecutará sobre el cliente (dispositivo móvil) y permitirá realizar todas las funciones del visualizador de reportes del ambiente de escritorio. (Autenticación, obtención de reportes y filtros dinámicos sobre los datos de reportes). Se compilará dicho paquete con el *framework* Phonegap, lo que permitirá obtener aplicaciones instalables en el dispositivo cliente de acuerdo a la tecnología de su plataforma.

### 4.2.1 Selección y análisis de la metodología de desarrollo.

Desde el punto de vista de Bohem y Turner (15), a partir de su matriz sobre el uso de metodologías ágiles o clásicas para el desarrollo de software, se puede fundamentar la metodología adecuada para la solución propuesta..

Criterio	Ágil	Clásica	Solución
<b>Objetivos primarios</b>	Rápidos, respuesta al cambio	Estabilidad, alta garantía	La solución debe ser versátil ante los cambios de arquitectura del servidor
<b>Tamaño</b>	Pequeño proyectos y equipos de desarrollo	Grandes equipos y proyectos	Proyecto de pequeño tamaño.
<b>Entorno</b>	Cambiante	Estable	Ambiente cambiante, dependiente del servidor.
<b>Planeación y control</b>	Planes y control cualitativo	Planes documentados, control cuantitativo	Control cualitativo a partir de resultado de las fases.

<b>Comunicación</b>	Conocimiento interpersonal	Conocimiento explícito documentado	Conocimiento implícito y personal
<b>Requerimientos</b>	Historias informales y casos de prueba	Formalizados, medibles y evolutivos	Historias informales basados en pruebas y experiencias de usuario
<b>Desarrollo</b>	Diseño simple, incrementos cortos	Diseño extensivo, incrementos largos	Desarrollo simple
<b>Prueba</b>	Casos de prueba	Pruebas documentadas	Casos de prueba para las historias de usuario

Tabla 1.1 Criterios de elección de entre metodologías tradicionales y ágiles

Estas y otras cuestiones, como las características del cliente y la cultura integral del equipo de desarrollo en función del proyecto, evidencian desde el punto de vista de Bohem y Turner, que para el proyecto en desarrollo la metodología adecuada es una metodología ágil. Esto certifica, aunque el proyecto SCADA utiliza para su desarrollo la metodología RUP, que en la investigación a realizar es conveniente una metodología ligera, escogiéndose XP.

## 2.5 Exploración y planificación

### 2.5.1 La fase de exploración y planificación

En XP se define la exploración y planificación como la etapa en la que se plantean las necesidades a automatizar a partir de las historias de usuario. A la vez, el equipo de desarrollo debe familiarizarse con el problema y establecer las fases de desarrollo de la solución, estudiar las posibles herramientas y definir las candidatas utilizando pruebas, para luego conceptualizar las bases arquitectónicas del sistema. Esta fase es corta, aunque su duración considera el dominio del problema por el equipo de proyecto (31). En esta fase, se definen las historias de usuario, los planes de iteración y de entregas.

### 2.5.2 Historias de usuario

Una historia de usuario es una funcionalidad concreta que se escribe desde el punto de vista del cliente, y contiene los aspectos que permiten su desarrollo como funcionalidad atómica del sistema. La historia debe ser lo suficientemente simple y clara para que sea comprendida por el cliente y lo suficientemente completa para que pueda ser implementada por un desarrollador. Cuando se define una historia de usuario, el equipo

de desarrollo debe estar claro de que es una funcionalidad indivisible en el sistema, en caso contrario, se sugiere que se divide en dos o más historias, ya que éstas deben ser desarrolladas en ciclos cerrados y cortos (32).

Para la presente investigación las historias de usuario parten de las siguientes funcionalidades:

- 1 Autenticar usuarios
- 2 Visualizar reportes
- 3 Visualizar datos de reportes
- 4 Filtrar reporte
- 5 Salvar datos de reportes en PDF
- 6 Salvar datos de reportes en Excel
- 7 Salvar datos de reportes en HTML

Las historias de usuario son las siguientes:

<b>Historia de Usuario</b>	
<b>Número: 1</b>	<b>Usuario:</b> Todos
<b>Nombre historia:</b> Autenticar usuarios	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Baja
<b>Puntos estimados: 2</b>	<b>Iteración asignada: 1</b>
<b>Programador responsable:</b> Yaimari Nuñez Aldana	
<p><b>Descripción:</b> El usuario accede a la página de autenticación desde su dispositivo móvil y visualiza un formulario de autenticación con los campos usuario y contraseña. Los campos deben contener textos en vacío que indiquen su función. El usuario y contraseña debe validarse según los permisos del módulo de seguridad del SCADA “Guardián del Alba”. En caso de ser incorrecto la página debe mostrar los siguientes mensajes:</p> <ul style="list-style-type: none"> <li>• Usuario no registrado: “Usted no tiene permisos para acceder al sistema”</li> </ul>	

<ul style="list-style-type: none"> <li>• Contraseña incorrecta: “La contraseña introducida no es válida”</li> <li>• Otros errores: “Se han detectado errores durante la autenticación, intente de nuevo”</li> </ul>
<b>Observaciones:</b>

Tabla 2.1 Historia de Usuario: Autenticar usuario

Historia de Usuario	
<b>Número: 2</b>	<b>Usuario:</b> Usuario Autenticado
<b>Nombre historia:</b> Visualizar reportes	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Yaimari Nuñez Aldana	
<p><b>Descripción:</b> Una vez autenticado el usuario, debe dirigirse a una página donde podrá obtener directamente los reportes generados por el servidor. De los reportes se debe mostrar en una tabla el nombre y la descripción. El nombre debe tener un link que le permita al usuario ver los detalles del reporte.</p>	
<b>Observaciones:</b>	

Tabla 2.2 Historia de Usuario Visualizar reportes

Historia de Usuario	
<b>Número: 3</b>	<b>Usuario:</b> Usuario Autenticado
<b>Nombre historia:</b> Visualizar datos de reporte	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media

<b>Puntos estimados: 2</b>	<b>Iteración asignada: 1</b>
<b>Programador responsable:</b> Yaimari Nuñez Aldana	
<p><b>Descripción:</b> Cuando el usuario ha dado clic sobre un reporte el cliente debe enviar al servidor el ID del reporte para saber si este es parametrizado o no. Si el reporte es parametrizado se debe mostrar una ventana con los posibles tipos de filtros para los datos del reporte. Luego a partir de estos parámetros debe aparecerle una vista con todos los datos del reporte, la vista mostrará una tabla donde cada columna será para un dato del reporte según los datos enviados por el servidor.</p>	
<p><b>Observaciones:</b> Cuando se realice una petición para obtener un reporte, solo visualizarán el dispositivo móvil los primeros 500 valores del reporte.</p>	

Tabla 2.3 Historia de Usuario Visualizar datos de reporte

<b>Historia de Usuario</b>	
<b>Número: 4</b>	<b>Usuario:</b> Usuario Autenticado
<b>Nombre historia:</b> Filtrar reporte	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados: 1.5</b>	<b>Iteración asignada: 2</b>
<b>Programador responsable:</b> Yaimari Nuñez Aldana	
<p><b>Descripción:</b> En la vista de reportes, se tendrá un campo que permitirá introducir un valor o cadena. Cuando el usuario seleccione el valor podrá ejecutar un filtro que restrinja la tabla de datos de reporte a aquellos que contienen valores definidos.</p>	
<b>Observaciones:</b>	

Tabla 2.4 Historia de Usuario Filtrar datos de reporte

<b>Historia de Usuario</b>	
<b>Número: 5</b>	<b>Usuario:</b> Usuario Autenticado
<b>Nombre historia:</b> Salvar datos de reporte a HTML	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados: 1.5</b>	<b>Iteración asignada: 2</b>
<b>Programador responsable:</b> Yaimari Nuñez Aldana	
<p><b>Descripción:</b> El usuario tendrá un botón en la parte superior en la vista de datos de reporte que permitirá Salvar los datos a HTML. Esto significa que el usuario podrá guardar la página HTML completa con los datos del reporte.</p>	
<p><b>Observaciones:</b> Para exportar datos a HTML solo se considerarán las primeras 500 filas del reporte señalado.</p>	

Tabla 2.5 Historia de Usuario Salvar datos de reporte a HTML

<b>Historia de Usuario</b>	
<b>Número: 6</b>	<b>Usuario:</b> Usuario Autenticado
<b>Nombre historia:</b> Salvar datos de reporte a Excel	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados: 1.5</b>	<b>Iteración asignada: 2</b>
<b>Programador responsable:</b> Yaimari Nuñez Aldana	

<p><b>Descripción:</b> El usuario tendrá un botón en la parte superior en la vista de datos de reporte que permitirá exportar los datos a Excel. La tabla con la información de los reportes debe descargarse en formato Excel en un directorio Descargas dentro del paquete del cliente.</p>
<p><b>Observaciones:</b> Para exportar datos a Excel solo se considerarán las primeras 500 filas del reporte señalado.</p>

Tabla 2.6 Historia de Usuario Salvar datos de reporte a Excel

Historia de Usuario	
<b>Número: 7</b>	<b>Usuario:</b> Usuario Autenticado
<b>Nombre historia:</b> Salvar datos de reporte a PDF	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 1.5	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Yaimari Nuñez Aldana	
<p><b>Descripción:</b> El usuario tendrá un botón en la parte superior en la vista de datos de reporte que permitirá exportar los datos a PDF; La tabla con la información de los reportes debe descargarse en formato PDF en un directorio Descargas dentro del paquete del cliente.</p>	
<p><b>Observaciones:</b> Para exportar datos a PDF solo se considerarán las primeras 500 filas del reporte señalado.</p>	

Tabla 2.7 Historia de Usuario Salvar datos de reporte a PDF

### 2.5.3 Requisitos no Funcionales

Un requisito no funcional es una propiedad o argumento que hace que el producto de software sea más atractivo y aceptado. Por lo general no define el éxito del producto, pero tiene gran impacto en su apreciación y evaluación por el cliente, llegando a éste primero que los valores funcionales. Concretamente, son propiedades y cualidades que el producto debe cumplir. Para el visualizador de reportes del SCADA “Guardián del Alba” para dispositivos móviles se definen los siguientes requisitos funcionales.

### **Usabilidad**

La aplicación será utilizada por las personas autorizadas del módulo de seguridad del SCADA “Guardián del Alba”. El usuario no tendrá necesariamente tener conocimientos avanzados de informática, pero sí de los reportes diseñados para el sistema y conocer la lógica de comprensión de los datos asociados al reporte, siendo necesario que sea personal familiarizado con la supervisión y control del proceso que sirve como escenario. Las funciones elementales como actualizar y filtrar estarán visibles al usuario de forma relevante haciendo más fácil su reconocimiento e identificación.

### **Seguridad**

El sistema responderá a los requerimientos de seguridad para módulos y aplicaciones del SCADA “Guardián del Alba” cumpliendo con la integridad, disponibilidad y confidencialidad de la información. El sistema estará embebidos en los dispositivos.

### **Interfaces externas**

Las interfaces externas deberán adaptarse a las resoluciones de los dispositivos (*responsive*). Contará con mínimas distracciones, colores monocromáticos y los elementos de acceso deben ser visibles y relevantes para el usuario. Las tablas y paginaciones deben ser visibles y la tipografía regular de la familia *Sans Serif*.

### **Rendimiento**

La aplicación debe ofrecer un rendimiento adecuado de hardware y la utilización de software en dependencia de las tecnologías seleccionadas.

Hardware: 32 Mb de Memoria, 28 Mb de Disco, 512 Kb de Procesamiento.

Software: Sistema con soporte para plataformas Android 4.4.1 o superior.

#### **2.5.4 Estimación de esfuerzo por historias de usuario**

La estimación de esfuerzo por historias de usuario permite al equipo de desarrollo determinar las fuerzas necesarias para construir la aplicación en el tiempo definido, identificando las cargas de trabajo de los desarrolladores y asegurando una visión del calendario para confeccionar los planes de entrega. A partir de los puntos estimados de las historias de usuario definidas para el visualizador de reportes a desarrollar se tiene la siguiente distribución de esfuerzos:

Historia de usuario	Esfuerzo en semanas
Autenticar usuario	2
Visualizar reportes	2
Visualizar datos de reportes	2
Filtrar reporte	1.5
Salvar datos de reporte a HTML	1.5
Salvar datos de reporte a Excel	1.5
Salvar datos de reporte a PDF	1.5
<b>Total</b>	<b>12</b>

Tabla 2. 8 Estimación de esfuerzo a partir de historias de usuario

### 2.5.5 Plan de iteraciones

El plan de iteraciones de XP permite describir las fases en las que el equipo divide el desarrollo de la solución. Siguiendo el paradigma de desarrollo iterativo e incremental, se asegura una organización del personal de proyecto para responder a las tareas de desarrollo, realizando un plan rector que se asume como documento de planificación del proyecto (33). La división de las historias por cada fase depende de su nivel de complejidad y las dependencias entre unas y otras. En este caso, el desarrollo se dividió en dos fases.

**Iteración No. 1:** Se desarrollarán las historias que dan entrada al proceso o que son necesarias para la visualización de reportes.

**Iteración No. 2:** Contiene las historias con el objetivo de filtros o exportaciones que dependen de las funcionalidades de entrada o representación de datos.

Iteración	Duración	
Iteración No. 1	2	6

	2	
	2	
<b>Iteración No. 2</b>	1.5	6
	1.5	
	1.5	
	1.5	

Tabla 2. 9. Plan de Iteraciones

### 2.5.6 Plan de entregas

El plan de entrega es el calendario de salida de terminación de cada fase, por lo que debe generar artefactos o segmentos del sistema entregables al sistema. Para el caso de la presente investigación se presenta el siguiente plan de entregas:

Iteración	Iteración No. 1	Iteración No. 2
<b>Cantidad de historias de usuario</b>	3	5
<b>Fechas de entrega</b>	4 de Abril de 2014	9 de Mayo de 2014

Tabla 2.10 Plan de entregas

## 2.6 Diseño de la solución propuesta

El diseño de la solución se ajustará al proceso de visualización de reportes del ambiente de escritorio del SCADA “Guardián del Alba”. Se desarrollará una aplicación que opere sobre el cliente (dispositivo móvil) y permita a través de la conexión con el servidor visualizar la información requerida. Para ello se utiliza el protocolo de comunicación WebSocket. Los detalles de la aplicación se podrán ver a partir de su arquitectura.

### 2.6.1 Arquitectura

La arquitectura propuesta se divide en dos componentes fundamentales: de un lado el paquete a desarrollar, compuesto por una serie de configuraciones escritas en HTML; JavaScript y CSS, que se ejecutan sobre el cliente y constituyen el visualizador de reportes. Este paquete se compilará para obtener una aplicación nativa para las familias de dispositivos móviles y se comunicará mediante WebSocket con el servidor, quien interpreta los mensajes del SCADA para proveer información a los terceros agentes que demandan información.



Figura 2.1 Vista de arquitectura entre el cliente y el servidor

La aplicación cliente, generada con la plataforma Phonegap, contendrá los siguientes componentes:

**Reglas CSS:** Las reglas CSS se usarán para dar estilos apropiados a los componentes a utilizar dentro de las vistas. Estas reglas serán responsivas, lo que permitirá a la aplicación adaptarse a la resolución de cualquier dispositivo desde el cual se acceda a los reportes. Los archivos CSS estarán encapsulados en la carpeta CSS del paquete. Como paquete de reglas para los estilos se utilizará Bootstrap, reconocido como un *framework* del lado del cliente para el desarrollo de aplicaciones web responsivas basado en las técnicas de HTML5 y CCS3 (34).

**Funciones JavaScript:** Para la ejecución de funciones del negocio y para la visualización a partir de operaciones se utilizará el lenguaje JavaScript. Se construirán a partir de dicho lenguaje las acciones de conexión, recolección de datos y configuración de la página, además de ser utilizado como una forma de interactuar con el usuario. Los archivos JavaScript estarán dentro del directorio JS del paquete propuesto. Para las configuraciones JS se utilizará jQuery. Esta es una librería de JavaScript que permite simplificar

la forma de trabajar con el documento HTML, manipular el árbol *DOM*<sup>3</sup> (siglas del inglés *Document Object Model* o Modelo de Objetos del Documento en español), manejar eventos y agregar interacción utilizando la técnica AJAX<sup>4</sup> (siglas del inglés *Asynchronous JavaScript And XML*).

**Página HTML:** Finalmente, el usuario interactuará con páginas HTML de donde se cargarán las configuraciones de JavaScript y CSS. Las páginas HTML serán varias y estarán en la raíz del paquete.

Toda la aplicación se desarrolla sobre una sola página web. Esta estructura permite utilizar una sola conexión de WebSocket sin necesidad de abrir y cerrar dicha conexión por cada proceso de la aplicación. La lógica para manejar las diferentes vistas de proceso se realizan con JQuery, dando flexibilidad y dinamismo a todo el sistema. Una vista del proceso de autenticación puede verse detalladamente en la imagen a continuación.



Figura 2.2 Vista completa del componente de autenticación del cliente

La simplicidad de utilizar un paquete de configuraciones web compilado mediante Phonegap ofrece las siguientes ventajas a la solución propuesta:

<sup>3</sup> DOM: Interfaz programable que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

<sup>4</sup> AJAX: Técnica de desarrollo web para crear aplicaciones interactivas (42).

- 1 Todo el código se ejecuta del lado del cliente, dando al usuario la posibilidad de dedicar poco espacio a la aplicación sin depender de recursos externos ni servicios adicionales para su funcionamiento.
- 2 Garantía de seguridad personal. La aplicación dentro de los dispositivos personales de operadores y ejecutivos, elimina las brechas de seguridad presentes en los servicios publicados en la web, siendo necesario enfocarse solamente en los aspectos de disponibilidad, confidencialidad e integridad del servidor.
- 3 Tasa de envío de datos baja. Con todas las configuraciones del lado del cliente no es necesario descargar datos de representación desde el servidor, haciendo más rápida la comunicación y permitiendo mayor rendimiento.
- 4 Poco consumo de recursos. Al ser una aplicación ligera el consumo de recursos disminuye, pudiéndose ejecutar en dispositivos de bajas prestaciones que soporten las requeridas de la plataforma.
- 5 Interpretación según la lógica del dispositivo cliente. Al obtenerse una aplicación nativa para el dispositivo móvil, la instalación, manipulación y gestión de recursos se hace más eficiente, pudiéndose vincular con los elementos físicos y lógicos del dispositivo (cámara, sensores, lista de contactos, etc.).

### **2.6.2 Interacción cliente servidor**

La comunicación con el servidor se realiza utilizando el formato definido para los terceros agentes que se comuniquen con el servidor web del SCADA “Guardián del Alba”. Este formato, estandarizado por el proyecto, permite obtener datos en forma de JSON, que pueden ser decodificados por el cliente para interpretarlos en una lógica natural y fácil de representar desde el lenguaje JavaScript. Como JSON es un formato ligero que puede ser utilizado bajo lenguajes C++, JavaScript, C#, Perl, Python y otros, ofrece la ventaja de cruzar varias capas de servicios cliente-servidor y utilizarse indistintamente en cada una de ellas.

Para cada tipo de petición al servidor, se utiliza una notación JSON diferente, obteniéndose en algunos casos una respuesta en el mismo formato y en otros, una vista totalmente construida para ser representada al usuario. A continuación se define un ejemplo de la utilización de JSON que es el proceso de autenticación de usuario.

Durante la autenticación el cliente envía una cadena JSON codificada en el siguiente formato:

```
[
  2,
  "7DK6TdN4wLiUJgNM",
  "/security/authenticate",
  {
    "user": "elpidio",
    "password": "elpidio123"
  }
]
```

Figura 2.3 Petición de Autenticación del Cliente al servidor

En este caso, la cadena contiene un identificador del tipo de llamada y otro para identificar la petición. La ruta de ejecución y un objeto compuesto por el nombre de usuario y contraseña. El servidor es capaz de decodificar la cadena y analizar sus atributos, identificando la necesidad de autenticación y procesando los datos en una capa específica para dicha tarea. Una vez procesada la petición envía una respuesta en el mismo formato:

```
[
  3,
  "7DK6TdN4wLiUJgNM",
  "/security/authenticate",
  {
    "userid": "45217890",
    "username": "Elpidio Valdes",
    "remainingtime": 20891245
  }
]
```

Figura 2.4 Respuesta de autenticación del servidor al cliente

La respuesta es interpretada por el cliente, y notifica su resultado a través de JQuery. En caso de error de autenticación o campos incorrectos, la cadena enviada contiene dentro del objeto, una dupla nombre/valor que indica el tipo de error y el mensaje, que es decodificado y mostrado al usuario.

```
[
  4,
  "7DK6TdN4wLiUJgNM",
  "/security/authenticate/errors",
  ""
  {
    "code": 751,
    "type": "error",
    "text": "Usuario o contrase\xfla incorrectos"
  }
]
```

Figura 2.5 Error de autenticación enviado por el servidor

### Utilización de vistas generadas por el servidor

Una vista generada es una colección de datos que el cliente puede utilizar directamente sin realizar transformaciones ni construir elementos dentro del HTML. Uno de estos casos es durante la obtención de los datos del reporte.

Cuando el cliente solicita los datos de reporte, lo hace mediante un formato JSON que contiene el identificador del reporte que se desea visualizar. El servidor procesa la petición identificando la solicitud del usuario, obtiene desde la base de datos la información del reporte ejecutando una consulta en el lenguaje declarativo de acceso a datos necesario, y construye una vista HTML que contiene toda la información a visualizar. Esta vista es enviada al cliente que solo tiene que representarla al usuario.

Esta variante permite obtener vistas construidas desde el propio servidor, liberando al cliente de esas responsabilidades, pudiéndose obtener vistas de diferentes tipos, como gráficos, tablas personalizadas y componentes dinámicos, que solamente tienen que cumplir con la lógica y formato de envío.

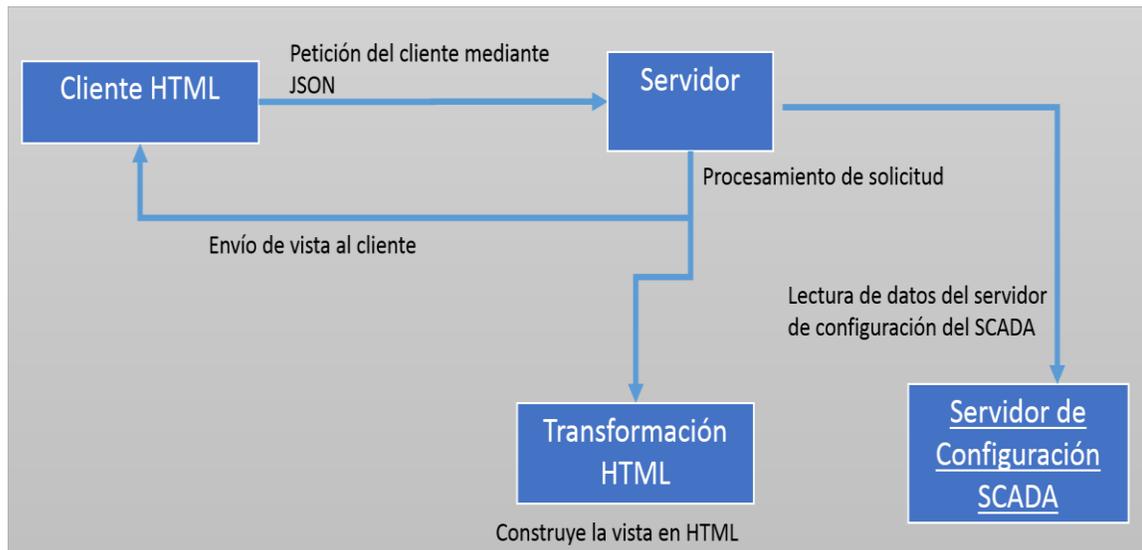


Figura 2.6. Proceso de solicitud, construcción y envío de vistas

### Procesamiento en el servidor

El servidor tiene un área de procesamiento que se encarga de determinar la naturaleza de las peticiones emitidas por el cliente. Estas peticiones son dirigidas a las clases especializadas, que realizan las gestiones de autenticación, lectura de datos o transformación. Una vez obtenida la información, es enviada nuevamente al cliente mediante el protocolo WebSocket.

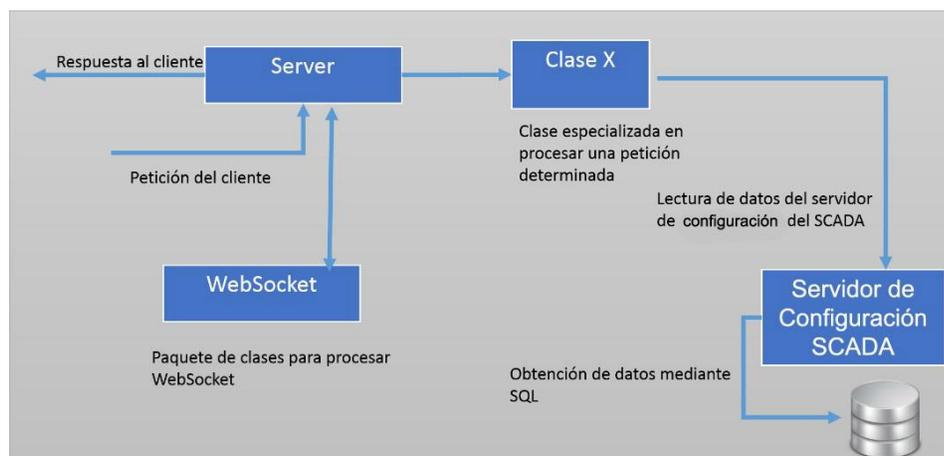


Figura 2.7 Mecanismo de procesamiento del servidor

### Vista completa de proceso cliente servidor

Para tener una descripción del proceso completo, se puede visualizar la interacción entre cliente-servidor en el siguiente diagrama.

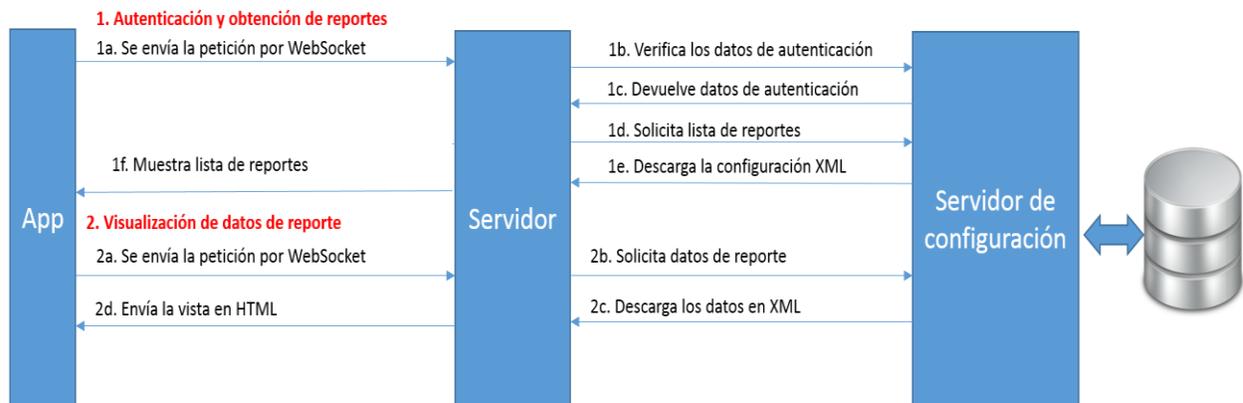


Figura 2.8 Secuencia para la solicitud de reportes

#### 2.6.4 Mapeo de datos de parametrización

Dentro de la lógica del negocio a considerar, existe la lectura y procesamiento de los reportes parametrizados. Un reporte parametrizado es aquel que puede ser filtrado a partir de campos específicos. Cuando el cliente solicita los datos de un reporte de este tipo, se deben evaluar los datos de parametrización para a partir de un formulario, introducir los filtros adecuados. Para dar cumplimiento a esta tarea, se desarrolló un componente JavaScript que permite la configuración de los tipos de datos contenidos en los filtros. Este componente tiene una función base que utiliza la configuración básica de HTML para construir dinámicamente campos dentro del formulario de filtros a partir de los parámetros obtenidos desde el servidor. El archivo contiene los tipos de campos del formulario en un arreglo, siendo posible adicionar o modificar cualquiera de éstos, añadiendo dinamismo a la aplicación y permitiendo que se puedan obtener cualquier tipo de parámetros en los filtros.

Con este mecanismo, la manipulación de reportes parametrizados es totalmente configurable y el cliente puede ser adaptado a cualquier requerimiento del servidor, definiéndose automáticamente los estilos y características de los campos del formulario de filtros utilizando las ventajas de HTML5.

### 2.6.5 Tarjetas CRC del lado del servidor

La metodología XP propone las tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración) para realizar el diseño de un sistema partiendo de la representación de objetos, las funcionalidades que deben cumplir éstos y los colaboradores que permiten que dichas funcionalidades se cumplan.

El modelo usado para la confección de las tarjetas CRC es el propuesto por la metodología.

**Clase:** Entidad que representa una clase en el sistema.

**Responsabilidad:** Son las operaciones que realizará la clase.

**Colaboradores:** Clases con las que se relaciona.

**Descripción:** Descripción de la responsabilidad de la clase.

A partir de la arquitectura propuesta, se detallan las tarjetas de las principales clases del servidor que interactúan con la aplicación propuesta.

<b>Clase: Server()</b>
<p><b>Responsabilidad:</b></p> <p>processNewConnection ()</p> <p>processMessage ()</p> <p>processPong ()</p> <p>socketDisconnected ()</p>
<p><b>Colaboradores:</b></p> <p>Login ()</p> <p>Report()</p> <p>AgentTrend ()</p>

AgentConfiguration () PointGenerator()
<b>Descripción: La clase Server es la interfaz fundamental del servidor encargada de procesar las peticiones del cliente.</b>

Tabla 2.11 Tarjeta CRC Clase Server

<b>Clase: ReportGenerator()</b>
<b>Responsabilidad:</b> generate () generateAReport () generateAShowReport ()
<b>Colaboradores:</b> XML () Report() QJSDocument () QJsonArray () QJsonObject ()
<b>Descripción: Clase que permite la generación de reportes a partir de las configuraciones XML descargadas del servidor.</b>

Tabla 2.12 Tarjeta CRC Clase ReportGenerator

## 2.7 Conclusiones parciales

La argumentación de las consideraciones del negocio permitieron establecer las características del ambiente de ejecución del sistema propuesto, así como los modelos, técnicas y herramientas que

actualmente se utilizan para la visualización de reportes en el ambiente de escritorio del SCADA “Guardián del Alba”, obteniéndose los requisitos funcionales necesarios para dar respuesta al objetivo de la investigación. Una vez analizadas las metodologías de desarrollo de software, se logró seleccionar aquella que se ajusta a las particularidades de la solución deseada, asegurando el cumplimiento de los planes de entrega. La modelación de los objetivos en forma de historias de usuario contribuyó a obtener las funcionalidades del sistema además de conseguirse la planificación a través de un plan de iteración y de entrega, necesario para guiar al equipo de desarrollo durante el ciclo de vida del proyecto. Por último, se estudió la arquitectura, y la lógica de comunicación cliente – servidor, propiciando la comprensión de los aspectos técnicos a tener en cuenta para la solución a implementar.

## Capítulo 3 Implementación y Pruebas

En el presente capítulo se realiza una descripción de los artefactos de la etapa de implementación de la solución propuesta. Se detallan además los resultados de las pruebas a través de las tarjetas de aceptación formuladas por la metodología y de la corrección de los errores encontrados durante las iteraciones. Por último, se describe el despliegue de la aplicación para argumentar el cumplimiento de los objetivos propuestos.

### 3.1 Implementación

La etapa de implementación obedece al desarrollo de las funcionalidades derivadas de los requisitos y las historias de usuarios. En esta etapa son elaboradas las tareas de ingenierías que responden a las funcionalidades que implementan las historias de usuarios; a partir de las mismas se registrará todo el proceso de desarrollo (20).

#### 3.1.1 Tareas de ingeniería

Una tarea de ingeniería es una actividad programable que el desarrollador conoce que el sistema debe hacer (32). Las tareas son subdivisiones de las historias de usuario, son estimables, con un tiempo de implementación corto, y sus objetivos son resolver los hitos de las historias de usuario a la cual pertenece. Una historia de usuario debe tener al menos una tarea de ingeniería. Pueden existir también tareas de ingeniería técnicas, que aunque no responden directamente a una historia, son partes programables necesarias para que el sistema funcione.

A partir de las historias de usuario elaboradas en la fase de exploración y planeación se definieron las siguientes tareas de ingeniería:

Historia de usuario	No. Tarea	Tarea de Ingeniería
Autenticar Usuario	1	Validar datos de usuario
Visualizar reportes	2	Obtener configuración de reportes
	3	Construir vista de reportes

Visualizar datos de reporte	4	Enviar datos de reporte
	5	Obtener configuración del reporte
	6	Evaluar parametrización del reporte
	7	Configurar vista del reporte
	8	Construir vista del reporte
Filtrar reporte	9	Capturar parámetro de búsqueda
	10	Construir vista del reporte
Salvar datos de reporte en PDF	11	Obtener configuración del HTML
	12	Construir fichero PDF
Salvar datos de reporte en Excel	13	Obtener configuración del HTML
	14	Construir fichero Excel
Salvar datos de reporte en HTML	15	Obtener configuración del HTML
	16	Salvar vista HTML

Tabla 3.1 Tareas de ingeniería del sistema

La descripción de las tareas de ingeniería definidas para la primera iteración del ciclo de desarrollo según los artefactos propuestos por la metodología es la siguiente:

Tarea	
<b>Número tarea: 1</b>	<b>Número historia: 1</b>
<b>Nombre tarea: Validar datos de usuario</b>	

<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 7-02-2014	<b>Fecha fin:</b> 21-02-2014
<b>Programador responsable:</b> Yaimari Nuñez Aldana	
<b>Descripción:</b> Se inserta en un formulario de autenticación el usuario y la contraseña. El sistema envía los parámetros al servidor. Éste devuelve un mensaje que debe ser interpretado para comprobar si los parámetros son válidos. El mensaje debe mostrarse en una ventana al usuario.	

Tabla 3.2 Tarea de Ingeniería Validar datos de usuario

Tarea	
<b>Número tarea:</b> 2	<b>Número historia:</b> 2
<b>Nombre tarea:</b> Obtener configuración de reportes	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 24-02-2014	<b>Fecha fin:</b> 28-03-2014
<b>Programador responsable:</b> Yaimari Nuñez Aldana	
<b>Descripción:</b> Una vez validada la autenticación se solicita al servidor los datos de reportes utilizando un formato JSON. El servidor descarga los datos desde en archivo XML definido por el servidor de configuración del SCADA y envía los reportes en otra secuencia JSON que es decodificada por el cliente.	

Tabla 3.3 Tarea de ingeniería Obtener configuración de reportes

Tarea	
<b>Número tarea:</b> 3	<b>Número historia:</b> 2
<b>Nombre tarea:</b> Construir vista de reportes	
<b>Tipo de tarea :</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 3-03-2014	<b>Fecha fin:</b> 7-03-2014

<b>Programador responsable: Yaimari Nuñez Aldana</b>
<b>Descripción:</b> La secuencia JSON obtenida desde el servidor es decodificada para obtener la lista de reportes. Se construye una vista HTML con las etiquetas y estilos necesarios para mostrar la tabla de reportes. Se oculta utilizando JQuery la vista de autenticación y se muestra la tabla con los reportes en formato HTML.

Tabla 3.4 Tarea de ingeniería Construir vista de reportes

Tarea	
<b>Número tarea: 4</b>	<b>Número historia: 3</b>
<b>Nombre tarea: Enviar datos de reporte</b>	
<b>Tipo de tarea : Desarrollo</b>	<b>Puntos estimados: 0.5</b>
<b>Fecha inicio: 10-03-2014</b>	<b>Fecha fin: 12-03-2014</b>
<b>Programador responsable: Yaimari Nuñez Aldana</b>	
<b>Descripción:</b> Se toma desde la tabla el índice del reporte seleccionado y se construye una secuencia JSON que contiene además la cadena de texto que representa la acción solicitada. La secuencia JSON es enviada al servidor.	

Tabla 3.5 Tarea de ingeniería Enviar datos de reporte

Tarea	
<b>Número tarea: 5</b>	<b>Número historia: 3</b>
<b>Nombre tarea: Obtener configuración de reporte</b>	
<b>Tipo de tarea : Desarrollo</b>	<b>Puntos estimados: 0.5</b>
<b>Fecha inicio: 12-02-2014</b>	<b>Fecha fin: 17-03-2014</b>
<b>Programador responsable: Yaimari Nuñez Aldana</b>	

**Descripción:** Se obtiene desde el servidor una petición en formato JSON. La secuencia es decodificada para analizar si el reporte es parametrizado o no. Si no es parametrizado se procede a configurar la vista del reporte, en caso contrario se evalúan los datos de parametrización.

Tabla 3.6 Tarea de ingeniería Obtener configuración de reporte

Tarea	
<b>Número tarea: 6</b>	<b>Número historia: 3</b>
<b>Nombre tarea: Evaluar parametrización de reporte</b>	
<b>Tipo de tarea : Desarrollo</b>	<b>Puntos estimados: 0.4</b>
<b>Fecha inicio: 17-03-2014</b>	<b>Fecha fin: 20-03-2014</b>
<b>Programador responsable: Yaimari Nuñez Aldana</b>	
<b>Descripción:</b> En caso del reporte ser parametrizado se analizan los parámetros contenidos en la sentencia JSON enviada por el servidor. A partir del formato de los parámetros se construye un formulario de filtro utilizando un script para la generación dinámica de los campos. La información del formulario es utilizada para filtrar los datos del reporte contenido en el resto de la cadena.	

Tabla 3.7 Tarea de ingeniería Enviar datos de reporte

Tarea	
<b>Número tarea: 7</b>	<b>Número historia: 3</b>
<b>Nombre tarea: Configurar vista del reporte</b>	
<b>Tipo de tarea : Desarrollo</b>	<b>Puntos estimados: 0.4</b>
<b>Fecha inicio: 20-02-2014</b>	<b>Fecha fin: 24-03-2014</b>
<b>Programador responsable: Yaimari Nuñez Aldana</b>	

**Descripción:** Se obtiene los elementos del JSON que contienen los datos específicos para el reporte y se añaden a un arreglo que pueda ser manipulado por el cliente para la construcción del HTML. Se esconde mediante JQuery la vista que contiene la tabla de reportes.

Tabla 3.8 Tarea de ingeniería Configurar vista del reporte

Tarea	
<b>Número tarea: 8</b>	<b>Número historia: 3</b>
<b>Nombre tarea: Construir vista del reporte</b>	
<b>Tipo de tarea : Desarrollo</b>	<b>Puntos estimados: 0.2</b>
<b>Fecha inicio: 24-03-2014</b>	<b>Fecha fin: 25-03-2014</b>
<b>Programador responsable: Yaimari Nuñez Aldana</b>	
<b>Descripción:</b> Se adiciona la vista HTML obtenida con los datos a las etiquetas contenedoras en el cliente. Las etiquetas son mostradas utilizando JQuery para permitir la visualización total de la información del reporte solicitado.	

Tabla 3.9 Tarea de ingeniería Construir vista del reporte

### 3.1.2 Vista de despliegue

La vista de despliegue es un artefacto que ayuda a comprender el escenario real donde estará desplegado el sistema, a partir de los nodos o componentes que comprenden la solución completa (15). Aunque no es un artefacto propio de la metodología seleccionada, dada su importancia se toma para representar el despliegue del sistema propuesto. Nótese en la siguiente vista que la aplicación se conecta desde el cliente al Servidor mediante el protocolo WebSocket. El servidor hace uso del servidor de configuración del SCADA para tomar los datos y realizar las operaciones correspondientes.

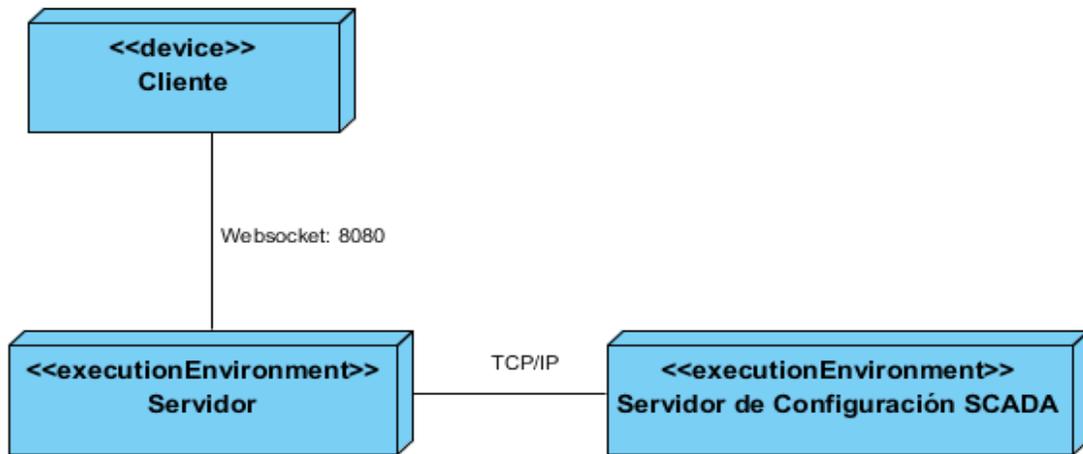


Tabla 3.1 Vista de despliegue de la solución

### 3.2 Pruebas

En la metodología XP la etapa de prueba tiene el objetivo de comprobar la terminación y satisfacción de los requisitos funcionales del sistema. Las pruebas se realizan al final de cada iteración y de ser conformes se avanza a la siguiente (31). A continuación se definen las principales causas de las No Conformidades detectadas durante las pruebas.

Iteración	Historia de Usuario	Causa
1	Autenticar usuario	Formato incorrecto del mensaje enviado por el servidor
	Visualizar reportes	Error en la decodificación de la sentencia JSON que contiene la configuración de reportes.
	Visualizar datos de reportes	Lectura incorrecta de los parámetros del reporte Construcción errónea del formulario de filtros.
2	Exportar datos a PDF	Mala definición de la dirección de salida del archivo.

Tabla 3.10 Resumen de errores durante las iteraciones del ciclo de desarrollo.

Un resumen gráfico del número exacto de errores es el siguiente

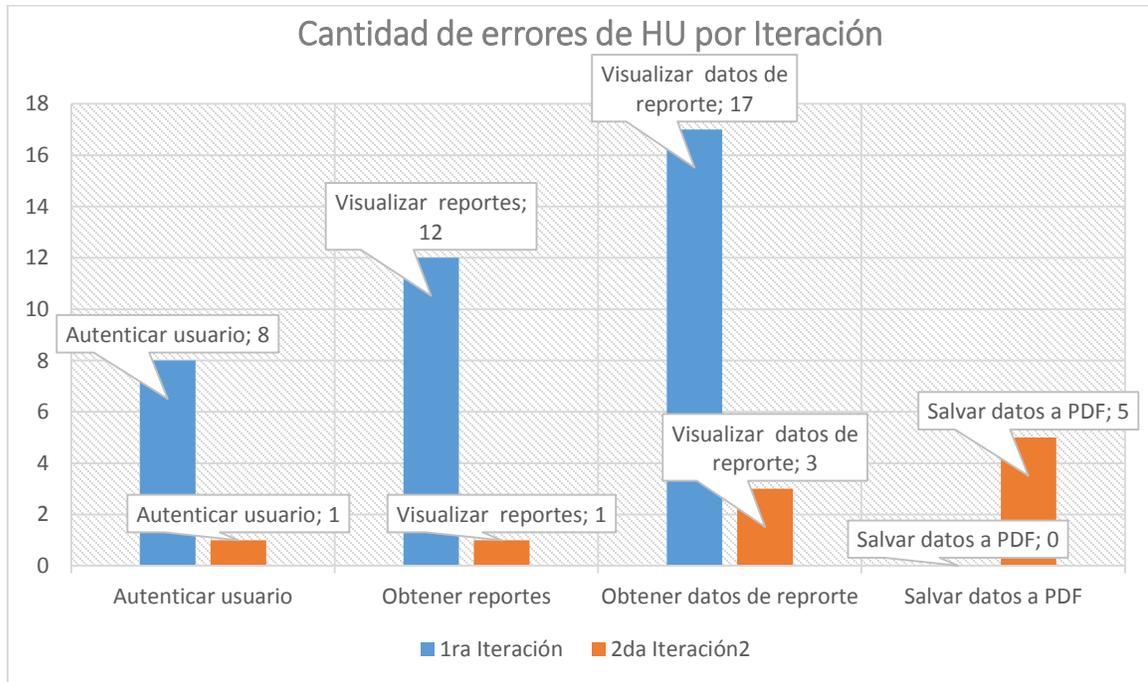


Figura 3.1 Cantidad de errores de Historia de Usuario por iteración

En la integración de la aplicación cliente con el servidor al terminar la segunda iteración se logró la corrección de todos los errores, validando el cumplimiento de los requisitos funcionales definidos, dando cumplimiento a los objetivos propuestos para esta investigación. La descripción detallada de los resultados de las pruebas al terminar el ciclo de desarrollo pueden observarse a partir de los Casos de Prueba.

### 3.2.1 Casos de prueba

Un caso de prueba es una validación de la terminación o no de una historia de usuario. La metodología XP propone realizar las pruebas a partir de pruebas funcionales (entre las que se encuentran las pruebas de aceptación) y/o pruebas unitarias. La prueba unitarias son validaciones escritas desde el punto de vista del desarrollador, mientras que las pruebas funcionales son escritas desde el punto de vista del cliente, alcanzando mayor importancia (20).

Para las dos iteraciones definidas se realizaron ciclos de prueba que arrojaron algunos errores que fueron corregidos luego de la integración cliente-servidor. Un resumen de las pruebas puede observarse a continuación.

Dentro de las pruebas funcionales las pruebas de aceptación son las de mayor relevancia, ya que constituyen la afirmación del cliente de que la historia está totalmente terminada. Una prueba de aceptación determina que tanto el requisito funcional como aquellos no funcionales involucrados en la historia de usuario fueron determinados como satisfactorios. A partir de las historias de usuario que respondían a los requisitos funcionales, se detallaron los siguientes casos de pruebas de aceptación para validar la culminación de la solución propuesta.

<b>Caso de Prueba</b>	
<b>Código:</b> P1	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Comprobar la autenticación de usuarios	
<b>Descripción:</b> A partir del formulario de autenticación se introducen los pares usuario/contraseña, que la aplicación debe enviar por JSON al servidor para su validación. La aplicación debe mostrar el mensaje de satisfacción o error y proceder a la siguiente operación.	
<b>Condiciones de ejecución:</b>	
<ol style="list-style-type: none"> <li>1 La aplicación debe contener la línea de configuración con los parámetros de conexión (dirección IP y puerto de conexión WebSocket).</li> <li>2 El servidor debe estar en ejecución con la misma configuración del cliente.</li> </ol>	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El usuario ejecuta la aplicación desde el dispositivo cliente.</li> <li>2. Introduce los datos de autenticación.</li> <li>3. El sistema envía los datos al servidor mediante WebSocket.</li> <li>4. El servidor envía la respuesta codificada en JSON.</li> <li>5. El cliente visualiza el mensaje de autenticación (error o éxito).</li> </ol>	
<b>Resultado esperado:</b> Los usuarios con accesos al sistema visualizan la lista de reportes.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

Tabla 3.10 Tarjeta de aceptación Autenticar Usuario

Caso de Prueba	
<b>Código:</b> P2	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Comprobar visualización de reportes	
<b>Descripción:</b> Una vez autenticado el usuario, se debe visualizar la lista de reportes a partir de la configuración descargada desde el servidor.	
<b>Condiciones de ejecución:</b> Debe existir una conexión abierta con el servidor a partir de la autenticación de usuario.	
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se solicita la configuración de reportes al servidor.</li> <li>2. El servidor envía los reportes en formato JSON.</li> <li>3. La sentencia JSON es decodificada para obtener la lista de reportes.</li> <li>4. Se configura la vista creando la tabla HTML con la lista de reportes.</li> <li>5. Se esconde el formulario de autenticación.</li> <li>6. Se visualiza la lista de reportes.</li> </ol>	
<b>Resultado esperado:</b> Los usuarios con accesos al sistema visualizan la lista de reportes.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

Tabla 3.11 Tarjeta de aceptación Visualizar reportes

Caso de Prueba	
<b>Código:</b> P3	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Comprobar visualización de datos de reportes	
<b>Descripción:</b> A partir de la lista de reportes el usuario escoge aquel que desea visualizar. El sistema obtiene la configuración del reporte desde el servidor y permite filtrarlo si es parametrizado. Al final muestra al usuario una tabla con los datos del reporte.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• Debe existir una conexión abierta con el servidor a partir de la autenticación de usuario.</li> <li>• Debe escogerse uno de los reportes de la lista.</li> </ul>	
<b>Entrada/Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se envía el índice del reporte al servidor.</li> </ol>	

<ol style="list-style-type: none"> <li>2. El servidor descarga la configuración del reporte desde el servidor de configuración del SCADA.</li> <li>3. El servidor evalúa si el reporte es parametrizado o no.</li> <li>4. Envía la configuración del reporte al cliente.             <ol style="list-style-type: none"> <li>a. Para reportes parametrizados.                 <ol style="list-style-type: none"> <li>i. Envía los parámetros al cliente.</li> <li>ii. El cliente construye un formulario de filtros.</li> <li>iii. El usuario introduce los datos de filtros.</li> <li>iv. Se restringen a partir de los parámetros.</li> </ol> </li> </ol> </li> <li>5. Se configura la vista de datos del reporte.</li> <li>6. Se muestra la lista de datos del reporte al usuario.</li> </ol>
<p><b>Resultado esperado:</b> Los usuarios acceden a los datos específicos del reporte.</p>
<p><b>Evaluación de la prueba:</b> Prueba satisfactoria</p>

Tabla 3.12 Tarjeta de aceptación Visualizar datos de reportes

<b>Caso de Prueba</b>	
<b>Código:</b> P4	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Comprobar Filtrar reporte	
<b>Descripción:</b> Cuando el usuario introduzca una cadena de texto en un campo en la parte superior de la vista de reportes para ejecutar un filtro que restrinja la tabla de datos de reporte a aquellos que contienen valores definidos.	
<p><b>Condiciones de ejecución:</b></p> <ul style="list-style-type: none"> <li>• Debe existir una conexión abierta con el servidor a partir de la autenticación de usuario.</li> <li>• Debe estar mostrada la lista de reportes.</li> </ul>	
<p><b>Entrada/Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Se introduce una cadena de texto en el campo de búsqueda.</li> <li>2. El sistema restringe la tabla a las filas cuyo nombre del reporte coincida parte del texto introducido.</li> <li>3. Se muestra al usuario la tabla restringida.</li> </ol>	
<p><b>Resultado esperado:</b> Los usuarios obtienen una vista de reportes con aquellos cuyo nombre coincide con parte del texto introducido en el campo de búsqueda.</p>	
<p><b>Evaluación de la prueba:</b> Prueba satisfactoria</p>	

Tabla 3.13 Tarjeta de aceptación Filtrar reporte

<b>Caso de Prueba</b>	
<b>Código:</b> P5	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Comprobar Salvar datos de reporte a PDF	
<b>Descripción:</b> El usuario da clic en el botón PDF y el sistema guarda la vista actual como una página PDF en la raíz del dispositivo.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• Debe existir una conexión abierta con el servidor a partir de la autenticación de usuario.</li> <li>• Debe estar mostrada la lista de datos de reportes.</li> </ul>	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El usuario hace clic en el botón Salvar como PDF.</li> <li>2. El sistema obtiene la vista HTML de los reportes.</li> <li>3. El sistema salva la vista en PDF en la raíz del dispositivo.</li> </ol>	
<b>Resultado esperado:</b> La vista de los datos del reporte queda salvada en la raíz del dispositivo en formato PDF.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

Tabla 3.14 Tarjeta de aceptación Salvar datos como PDF.

<b>Caso de Prueba</b>	
<b>Código:</b> P6	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Comprobar Salvar datos de reporte a Excel	
<b>Descripción:</b> El usuario da clic en el botón Excel y el sistema guarda la vista actual como una página Excel en la raíz del dispositivo.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• Debe existir una conexión abierta con el servidor a partir de la autenticación de usuario.</li> <li>• Debe estar mostrada la lista de datos de reportes.</li> </ul>	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El usuario hace clic en el botón Salvar como Excel.</li> <li>2. El sistema obtiene la vista HTML de los reportes.</li> <li>3. El sistema salva la vista en Excel en la raíz del dispositivo.</li> </ol>	

<b>Resultado esperado:</b> La vista de los datos del reporte queda salvada en la raíz del dispositivo en formato Excel.
<b>Evaluación de la prueba:</b> Prueba satisfactoria

Tabla 3.15 Tarjeta de aceptación Salvar datos como Excel.

Caso de Prueba	
<b>Código:</b> P7	<b>Historia de Usuario:</b> 7
<b>Nombre:</b> Comprobar Salvar datos de reporte a HTML	
<b>Descripción:</b> El usuario da clic en el botón HTML y el sistema guarda la vista actual como una página HTML en la raíz del dispositivo.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• Debe existir una conexión abierta con el servidor a partir de la autenticación de usuario.</li> <li>• Debe estar mostrada la lista de datos de reportes.</li> </ul>	
<b>Entrada/Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El usuario hace clic en el botón Salvar como HTML.</li> <li>2. El sistema obtiene la vista HTML de los reportes.</li> <li>3. El sistema salva la vista en la raíz del dispositivo.</li> </ol>	
<b>Resultado esperado:</b> La vista de los datos del reporte queda salvada en la raíz del dispositivo en formato HTML.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

Tabla 3.16 Tarjeta de aceptación Salvar datos como HTML.

Caso de Prueba	
<b>Código:</b> P8	<b>Historia de Usuario:</b> 8
<b>Nombre:</b> Comprobar Cerrar sesión de usuario	
<b>Descripción:</b> El usuario hace clic en el botón cerrar sesión y el cliente cierra la conexión con la aplicación.	
<b>Condiciones de ejecución:</b>	
<ol style="list-style-type: none"> <li>3 Debe existir una conexión abierta con el servidor a partir de la autenticación de usuario.</li> </ol>	

<b>Entrada/Pasos de ejecución:</b>
1 El usuario hace clic en el botón Cerrar sesión. 2 El sistema cierra la conexión con el servidor.
<b>Resultado esperado:</b> El cliente se desconecta del servidor.
<b>Evaluación de la prueba:</b> Prueba satisfactoria

Tabla 3.17 Tarjeta de aceptación Cerrar sesión de usuario.

### 3.3 Conclusiones parciales

En el presente capítulo se hizo referencia a todo el proceso de implementación del visualizador de reportes propuesto, argumentado a partir de las tareas de ingeniería construidas por los desarrolladores para dar respuesta a las historias de usuario definidas por el cliente. Como parte del proceso de pruebas, se expusieron los errores obtenidos y su distribución por cada iteración, así como las Tarjetas de Aceptación de los Casos de Pruebas que validan el cumplimiento del ciclo de desarrollo y los objetivos de la investigación.

## Conclusiones

Culminado el desarrollo del visualizador de reportes del SCADA “Guardián del Alba” en dispositivos móviles, se arribó a una solución completa que responde a los objetivos planteados y asegura la representación de la información de los reportes diseñados en el ambiente de escritorio del sistema, arribándose a las siguientes conclusiones:

- 1 La elección de la tecnología web y su compilación como aplicaciones nativas para las plataformas móviles permitió una solución flexible que puede ser integrada a diferentes ambientes y utilizada a distintos niveles del negocio donde se despliegue el SCADA “Guardián del Alba”, aumentando su accesibilidad y capacidad de estudio del fenómeno que se automatiza.
- 2 La simplicidad del modelo propuesto y la utilización del protocolo WebSocket, aseguraron una compatibilidad con las demás soluciones web del SCADA “Guardián del Alba”, y propició un sistema extensible, comprensible y que responde a los formatos y estándares de comunicación del proyecto.
- 3 Con el sistema de visualización de reportes en dispositivos móviles, los supervisores y ejecutivos tienen acceso a la información histórica del sistema, lo que permitió a la capa estratégica de la administración un acceso adecuado a información sensible para la toma de decisiones desde sus dispositivos personales, contribuyendo a mejorar la eficiencia y eficacia en la supervisión y control de procesos industriales.

## Recomendaciones

Se proponen las siguientes recomendaciones:

1. Integrar a la herramienta otras características como la llamada directa a operadores y controladores del negocio a partir de la manipulación directa de la lógica del dispositivo móvil.
2. Extender la aplicación a otras plataformas móviles que permitan su utilización en dispositivos de gamas bajas, altas y medias para propiciar un mayor acceso a los reportes del SCADA “Guardián del Alba”.

## Bibliografía

1. **van der Graaf, Shenja.** *Information Communication Technologies and Emergin Business Strategies*. London, UK : Idea Group Inc., 2007. ISBN 1-59904-236-3.
2. **Duggan, Evan y Reichgelt, Han.** *Measuring Information Systems Delivery Quality*. Hershey : Idea Group Inc., 2006. ISBN 1-59140-859-8.
3. *Dispositivos Móviles*. **Guevara, Anaid.** 7, México : Universidad Autónoma de México, 6 de Agosto de 2010, Seguridad, Defensa Digital, págs. 07-08.
4. **Banerjee A.** Arquitectura móvil. *The Architecture Journal*. s.l. : 14: p. 43, 2011.
5. Comunidades. [En línea] [Citado el: 27 de 1 de 2014.] <http://comunidades.uci.cu/projects/dynamicreport..>
6. danysoft. [En línea] [Citado el: 24 de 1 de 2014.] <http://www.danysoft.com/bol/crystal10.htm>.
7. [En línea] [Citado el: 24 de 1 de 2014.] <http://www.codigoadicto.com/software-crystal-reports-14-beta-1-para-visual-studio-2010>.
8. **Desarrollo.** *Equipo de. PATSI GDR. Manual de usuario*. Habana, Cuba : s.n., 2010.
9. **Cruz, Lianet.** *Módulo de Reportes para el Generador Dinámico de Reportes v2.0*. Facultad 6. La Habana : Univesidad de las Ciencias Informáticas, 2012. pág. 61, Tesis.
10. fast Report. [En línea] [Citado el: 18 de 2 de 2014.] <http://www.fast-report.com/en/blog/42/show/> .
11. emdmobile. [En línea] [Citado el: 18 de 2 de 2014.] <http://exemplomedical.com/specialties/additionalsolutions/emdmobile/Reports.aspx>.
12. eMDMobile. [En línea] [Citado el: 18 de 2 de 2014.] <http://exemplomedical.com/specialties/additionalsolutions/eMDMobile.aspx>.
13. SAPBusinessOne. [En línea] [Citado el: 18 de 2 de 2014.] <http://nsbizsolutions.com/Products/SAPBusinessOne/Mobile.aspx>.
14. Jasperforge. [En línea] abril de 2012. [Citado el: 4 de 2 de 2014.] <http://jasperforge.org/>.
15. **Jacobson, Ivar, Booch, G. y Rumbaugh, J.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación, 2000. ISBN: 84-7829-036-2.
16. **Krcuten, P.** *The Rational Unified Process: An Introduction*. s.l. : Addison Wesly, 2004. ISBN: 978-032-1197-70-2.

17. **Probasco, L.** *The ten essentials of RUP. The essence of a effective development proceess.* Lexington : Rational Software, 2000.
18. **Addison Wesley object technology series.** *Softwre development for small team: A RUP - centric approach.* s.l. : Addison Wesley, 2004. ISBN: 978-0321-199-50-8.
19. **McConell, S.** *Professional Software Development.* s.l. : Addison Wesley, 2003. ISBN : 0-321-19367-9 .
20. **Beck.** *Extreme Programming Explained, embrace change.* s.l. : Pearson Education, 2012. ISBN: 978-8131-704-51-6.
21. **Baisley, B.** *Unified Mdeling Language: infraestructure.* s.l. : Pearson Education, 2003.
22. **Kyrnin, Jennifer.** *HTML5 Mobile Application Development in 24 Hours.* 2012.
23. **PÉREZ, J. E.** *Introducción a CSS.* s.l. : Edtion ed 241 p, 2004.
24. **W3C.** [En línea] [Citado el: 15 de 2 de 2014.] <http://www.w3.org/TR/websockets/>.
25. **NetBeans.** NetBeans. [En línea] 13. <http://netbeansaccesible.blogspot.com/>.
26. **Rodríguez, Aquilino.** *Sistemas SCADA.* Segunda. s.l. : MARCOMBO S.A, 2006. ISSN: 8426714188.
27. **Kant, Krishna.** *Computer-Based Industrial Control.* s.l. : PHI Learning Pvt. Ltd, 2004. 9788120311237.
28. **Mandado, Enrique, y otros, y otros.** *Autómatas programables y sistemas de automatización.* México : Instituto de Electrónica Aplicada, 2009. ISBN 978-607-7686-73-6.
29. **Munayev , Yury Victor.** *XStormClient, visualizador web para SCADA "Guardián del Alba".* La Habana : Universisad de las Ciencias Informáticas. Facultad 5, 2012.
30. [En línea] 2012. [Citado el: 2 de 2 de 2014.] <http://es.scribd.com/doc/52131154/telefonía-celular>.
31. **Bent, Kent y Fowler, Martin.** *Planning Extreme Programming.* s.l. : Addison Wesley, 2001. ISBN: 0-201-71091-9..
32. **Wallace, Doug y Ragget, Isobel.** *Extreme Programming for Web Projects.* s.l. : Addison Wesley, 2002. ISBN : 0-201-79427-6.
33. **Penadés, Patricio Letelier y M<sup>a</sup> Carmen.** *Métodologías ágiles para el desarrollo de software.* [En línea] 2009. <http://www.willydev.net/descargas/masyxp.pdf>.
34. **Otto, Mark y Thornton, Jacob.** *Bootstrap.* [En línea] 19 de Agosto de 2011. [Citado el: 2014 de Marzo de 26.] <http://www.getbootstrap.com/>.
35. *Dispositivos móviles.* [En línea] [Citado el: 1 de 22 de 2014.] [http://leo.ugr.es/J2ME/INTRO/intro\\_4.htm](http://leo.ugr.es/J2ME/INTRO/intro_4.htm).

36. [En línea] [Citado el: 9 de 2 de 2014.] <http://xmlrpc.scripting.com/spec.html>.
37. *Simple Object Access Protocol (SOAP) Tutorial*.
38. *Modelado, Diseño e Implementación de Servicios Web*. **Marsset, Rafael Navarro**. 2007.
39. **Llerena, MSc. Gloria María Guerrero**. Metodología de desarrollo de Software. Cuba : s.n., 2008.
40. **ISAÍAS CARRILLO PÉREZ, R. P. G., AURELIANO DAVID RODRÍGUEZ MARTÍN**. Metodología del desarrollo de software.2008. [En línea] [Citado el: 2 de 12 de 2013.]
41. **Fowler, Martin, Rice, David y Foemmel, Matthew**. *Patterns of Enterprise Application Architecture*. s.l. : Addison Wesley, 2002. ISBN : 0-321-12742-0 .
42. **Crane, D y Pascarello, E**. *Ajax in Action*. Greenwich : Manning Publications, 2010. ISBN: 1-932394-61-3.