

Universidad de las Ciencias Informáticas
Facultad 5



Título: Módulo de visualización web para el generador de informes del Proyecto
SCADA-GALBA.

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

Autor(es): Luisa Mora Pérez
Roberto Rodríguez Reyes

Tutor(es): Ing. Raudi Agdel Bacallao Sánchez
Ing. Bernardo Zaragoza Hijuelos

Consultante: Ing. Enrique Reyes Bermúdez
Ing. Ernesto Leyva Barrero

Ciudad de La Habana, junio del 2010.
“Año 52 de la Revolución”

Declaración de autoría

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 2010.

Firma del Autor: Luisa Mora Pérez

Firma del Autor: Roberto Rodríguez Reyes

Firma del Tutor: Ing. Raudi Bacallao Sánchez

Firma del Tutor: Ing. Bernardo Zaragoza Hijuelos

Datos de Contacto

Ing. Raudi Agdel Bacallao Sánchez

e-mail : rabacallao@uci.cu

Graduado de Ingeniería en Ciencias Informáticas en el 2008 con 2 años de experiencia en la especialidad.

Ing. Bernardo Zaragoza Hijuelos

e-mail : bzaragoza@uci.cu

Graduado de Ingeniería en Ciencias Informáticas en el 2008 con 2 años de experiencia en la especialidad.

Dedicatoría

Dedicatoria

De Luisa:

Dedico este triunfo:

A mi papá, por ser incansable durante estos cinco años.

A mi mamá, por ser única y por su apoyo en todo momento.

A mi tía y a mi abuela por toda la confianza que depositaron en mí.

De Roberto:

*A mi mamita, por tener el 1er lugar en todo,
por ser mi mayor motivo de alegría, por ser la fuente de toda
mi fuerza y por convertirme en un hombre de principios;
el amor que siento por ella es indescriptible.*

Agradecimientos

De Luisa:

- ✓ *A mi mamá por ser maravillosa.*
- ✓ *A mi papá por ser incansable en esta batalla.*
- ✓ *A mi tía por depositar toda su confianza en mí y apoyarme en todas mis decisiones.*
- ✓ *A mi abuela por cuidarme y querer siempre lo mejor para mí.*
- ✓ *A mi tío Rolando que aunque no está a mi lado hace mucho, siempre lo tendré presente, pues fue junto con mi tía Mirella quienes me educaron cuando era pequeña y me dieron todo el amor del mundo.*
- ✓ *A mi familia porque desde pequeña siempre me dieron mucho amor y por confiar siempre en mí.*
- ✓ *A Roberto por ser mi compañero de tesis.*
- ✓ *A mis tutores por toda su ayuda.*
- ✓ *A Carlos, Yadira Marín, Yusmi, kike, Ernesto, gracias a todos ustedes por su ayuda incondicional siempre que los necesité.*
- ✓ *A Wilfredo y Analeida por abrirme las puertas de su casa y quererme como a una hija.*
- ✓ *A profesoras como Zenaida, Idelkys y Zaida que contribuyeron en gran medida a este logro.*
- ✓ *A Mazán por siempre estar ahí cuando la he necesitado.*
- ✓ *A Yady por hacer suyos mis problemas y ayudarme siempre en todo lo que ha sido necesario, gracias por ser para mí como una hermana.*
- ✓ *A Yili por estar siempre empujándome a hacer las cosas y por ser mi amiga.*
- ✓ *A Yuniet por ser siempre mi alegría.*
- ✓ *A Ubalquis por ser mi amigo en todo momento y por todos los consejos que me ha dado.*
- ✓ *A Yendris y Amides, gracias por siempre aconsejarme y querer siempre lo mejor para mí.*
- ✓ *A David por enseñarme que todo lo que en verdad me proponga lo puedo lograr.*
- ✓ *A Dollys por confiar siempre en que todo va a salir bien.*
- ✓ *A Leticia por su amistad.*
- ✓ *A Yamila, Ailem y Diana Liss, gracias por ser mis amigas y estar siempre dispuestas a ayudar en todo lo que me ha hecho falta.*
- ✓ *A mi churry por estar a mi lado en estos últimos días.*

Agradecimientos

- ✓ *A Sardi porque todas sus boberías me contagian de alegría.*
- ✓ *A Yanay y Abigail por siempre darnos fuerzas.*
- ✓ *A mis compañeros de los grupos 5107, 5308 y 5505, gracias a todos ustedes por ser tan divinos, nunca los olvidaré y a todos los amigos que he cultivado durante estos cinco años.*
- ✓ *Al comandante, por su idea genial de crear la UCI y por siempre confiar en los jóvenes.*

De Roberto:

Quiero agradecer a toda mi familia por ser la esencia y el pilar de todo lo logrado por mí hasta hoy, ellos siempre han sido mi faro y les estaré eternamente agradecido.

- ✓ *A mi mamá que ha derrochado sobre mí todo su amor, sus consejos, su vida entera me la ha dado y siempre se ha mostrado feliz de hacerlo.*
- ✓ *A mi abuela “Popo” que ha sido mi otra mamá y mi eterna defensora.*
- ✓ *A mi papá por ser un digno ejemplo a seguir y estar siempre cuidándome.*
- ✓ *A Ismany por quererme como a un hijo.*
- ✓ *A mi novia Lianny que me ha apoyado en estos últimos meses y por ser tan comprensiva.*
- ✓ *A Luisa que siempre confió en mí y no dudó en acompañarme en esta difícil tarea.*
- ✓ *A nuestros papás tutores Raudi y Bernardo, su mérito es indiscutible.*
- ✓ *A Kike y a Ernesto que siempre tuvieron la mano extendida para nosotros.*
- ✓ *A Neimy por ser la hermanita que nunca tuve.*
- ✓ *A todo el clan del antiguo grupo 3 y del actual extraordinario grupo 5, que hicieron de esta carrera una experiencia única.*
- ✓ *A mis amigos, que se transformaron en hermanos: Sosa, Sardi, Yuniel, Yoel, Yendri; sepan que son el mejor piquete universitario.*
- ✓ *A Yanay que tanta confianza y valor me imprimió.*
- ✓ *Es imposible para mí en tan pocas líneas agradecer a todas las personas que quiero, la vida en su transcurrir dará muestras a todos de mi afecto y aprecio.*

Resumen

En los últimos años el crecimiento de las industrias ha conllevado a la extensión de las fábricas y maquinarias que las componen, lo que ha traído como consecuencia un aumento en la complejidad de los procesos industriales. Para lograr un control eficiente de estos procesos surgieron los sistemas de supervisión, control y adquisición de datos, conocidos como SCADA, que es un término tomado del Inglés, acrónimo de Supervisory Control and Data Acquisition.

Este trabajo surge a partir de la necesidad de emitir informes desde la web en el generador de informes del proyecto SCADA-GALBA (Guardián del Alba). Refleja una investigación sobre las tecnologías que más se utilizan en el ámbito de aplicaciones web, orientándola hacia la automática. El objetivo de esta investigación es obtener un módulo de visualización web capaz de representar y supervisar distintos tipos de gráficos y controles previamente diseñados mediante una aplicación confiable y a la vez amigable, que permita la interacción con los componentes representados y que sea accesible desde cualquier aplicación. Para cumplir esta meta se describe una solución partiendo del diseño de software y se detalla la implementación del sistema.

Palabras clave

Adquisición de datos, control, módulo, SCADA, supervisión, visualización.

Índice de Contenidos

INTRODUCCIÓN..... 1

CAPÍTULO 1..... 5

 Introducción 5

 1.1 Introducción a los sistemas SCADA-GALBA..... 5

 1.2 Módulos del SCADA-GALBA 7

 1.3 Generación de informes en los sistemas SCADA 8

 1.3.1 Generalidades sobre los generadores de informes 9

 1.4 Evaluación de generadores de informes web 10

 1.4.1.1 Stimulsoft Reports.Web 11

 1.4.1.2 XML Report Generator (XMLRG)..... 11

 1.4.1.3 ReportAllWeb 12

 Con esta aplicación se puede: 13

 1.4.2 Generadores de informes bajo los principios del Software Libre 15

 1.4.2.1 JasperReport 15

 1.5 Tendencias y tecnologías actuales a considerar 16

 1.5.1 Software Libre 16

 1.5.2 Ambiente de Desarrollo Integrado (IDE) 17

 1.5.2.1 Eclipse..... 17

 1.5.2.2 Code::Blocks..... 18

 1.5.3 Cliente Web 18

 1.5.4 Servidor Web 19

 1.5.5 Protocolo de comunicación HTTP 19

 1.5.6 Lenguajes de programación para la web 20

 1.5.6.1 Perl (Practical Extracting and Reporting Language) 20

 1.5.6.2 Lenguaje HTML..... 20

 1.6 Framework web 21

 1.6.1 C++ Web Toolkit..... 21

 1.7 Lenguajes de programación 23

 1.7.1 C++ 23

 1.8 Biblioteca CxxTest 24

 1.9 Biblioteca Pqxx 24

 1.10 Herramientas CASE 24

 1.10.1 Visual Paradigm 25

 1.10.2 IBM Rational Software Architect (RSA)..... 25

 1.11 Metodologías de Desarrollo de Software 26

 1.11.1 RUP..... 26

 1.12 Propuesta de solución..... 27

 1.13 Conclusiones 28

CAPÍTULO 2..... 29

 Introducción 29

 2.1 Arquitectura de la solución 29

 2.2 Paquetes de diseño arquitectónicamente significativos 31

 2.2.1 Paquete Base..... 32

 2.2.2 Paquete Graphics 33

 2.2.3 Paquete WT..... 33

 2.2.3.1 Paquete View 34

2.2.3.2	Paquete Draw.....	34
2.2.4	Paquete IO (Input/Output).....	34
2.2.5	Paquete <i>Exceptions</i>	35
2.2.6	Paquete Utils.....	36
2.2.7	Paquete Types.....	36
2.3	Patrones.....	37
2.4	Principales funcionalidades.....	41
2.4.1	Visualizar Informes.....	41
2.4.2	Exportar informes.....	42
2.5	Conclusiones.....	42
CAPÍTULO 3	43
Introducción.....		43
3.1	Pruebas.....	43
Los diferentes niveles de pruebas son:.....		43
3.1.1	¿Qué es una Prueba de Unidad?.....	43
3.2	Diseño de las Pruebas de Unidad.....	44
3.3	Pruebas de clases dentro del método “Visualizar Informe”.....	44
3.3.1	Clase TestEngine.....	44
3.3.2	Clase TestReport.....	46
3.3.3	Clase TestDocument.....	47
3.4	Pruebas a las clases dentro del método “Exportar Informe a CSV”.....	49
3.3.4	Clase ExportReport.....	49
3.5	Resumen de Pruebas.....	50
3.6	Conclusiones.....	50
CONCLUSIONES.....		51
RECOMENDACIONES.....		52
REFERENCIAS BIBLIOGRÁFICAS.....		53
BIBLIOGRAFÍA.....		55
ANEXOS.....		56
GLOSARIO.....		74

Índice de Figuras

Fig. 1. Esquema de un Sistema de Adquisición, Supervisión y Control de Datos. 6

Fig. 2. Estructura general de un generador de informes..... 10

Fig. 3. Report Designer in Web..... 11

Fig. 4. XML Report Generator..... 12

Fig. 5. Ejemplo de un reporte generado que tiene vínculos definidos (Drill down/up). 14

Fig. 6. Reporte exportado a Excel. 15

Fig. 7. Representación de la interfaz de la Aplicación Jasper Report Ever. 16

Fig. 8. Fases de RUP..... 27

Fig. 9: Diagrama del patrón arquitectónico Tres Capas. 30

Fig. 10. Diagrama de clases. Paquete Base. Ver descripción 32

Fig. 11. Diagrama de Clases. Paquete Graphics. Ver descripción 33

Fig. 12. Diagrama de clases. Paquete Draw. Ver descripción 34

Fig. 13. Diagrama de clases. Paquete IO. Ver descripción..... 35

Fig. 14. Diagrama de clases. Paquete Exceptions. Ver descripción..... 35

Fig. 15. Diagrama de clases. Paquete Utils. 36

Fig. 16. Diagrama de clases. Paquete Types..... 36

Fig. 17: Diagrama del patrón Singleton..... 38

Fig. 18: Diagrama del patrón Creador. 39

Fig. 19: Diagrama del patrón Controlador. 40

Fig. 20: Diagrama de clases Visualizar Informe..... 41

Introducción

Históricamente, la necesidad de aumentar los niveles de eficiencia, minimizar los costos y optimizar los procesos de producción, han resultado sumamente importantes para cualquier industria. Actualmente, uno de los mecanismos más importantes para dar solución a estas necesidades, es la automatización de los distintos procesos que componen la cadena de producción, por tal motivo la necesidad de automatizar operaciones resulta prioritaria para cualquier industria donde se hayan invertido importantes recursos.

El advenimiento de los nuevos avances tecnológicos trajo soluciones eficientes al problema. Las computadoras agregaron la capacidad de programar tareas para realizar funciones más complejas y, a su vez, se han ido perfeccionando y adaptando a las condiciones que imponen el paso de las Tecnologías de la Información y las Comunicaciones (TIC) en el mundo.

Un ejemplo de esto, son los Sistemas de Control Industrial (ICS), término general que encapsula a varios tipos de sistemas de control como los SCADA. Los primeros SCADA fueron diseñados para cumplir funciones específicas dentro de un proyecto determinado. Estos pueden proporcionar toda la información que se genera en el proceso productivo a diversos usuarios relacionados directamente con el control y supervisión de dichos procesos así como a otros pertenecientes a niveles superiores dentro y fuera de la empresa como pueden ser control de calidad, supervisión o mantenimiento. En las bases de datos históricas del SCADA se registran los valores instantáneos de las variables de proceso, los eventos y la configuración del sistema, entre otras. Para muchos usuarios del SCADA, interesados en comportamientos más globales del sistema como estadísticas del proceso, detección de fugas y deterioro de indicadores productivos, esta vasta información carece de valor, a menos que sea procesada y presentada de forma más adecuada. En este punto surge la necesidad de emitir informes que consoliden la información adquirida para entregarla en un formato determinado, de tal manera que sea útil al usuario que va dirigida.

En la Universidad de la Ciencias Informáticas existen proyectos productivos cuyo objetivo fundamental es lograr la automatización de empresas e instituciones tanto del país como en el exterior. En el Centro de Informática Industrial, de la Facultad 5, se lleva a cabo el Proyecto SCADA-GALBA. Este proyecto cuenta con mecanismos para la generación de informes. Sin duda la generación de informes ha constituido una tarea primordial y crítica en muchas industrias, ya que se dedican esfuerzos considerables para realizar operaciones básicas como son agrupaciones, saltos de página, y conseguir que todo salga correctamente

impreso. La generación de informes posibilita procesar de forma automática los datos existentes en las bases de datos y generar un documento sobre la base de un esquema de informe prediseñado, el cual se puede visualizar, imprimir o exportar a algún formato en específico. Para los usuarios ajenos a la empresa, interesados en acceder a servicios integrados en la generación de informes, les es muy complejo acceder a estos; a menos que accedan directo a la aplicación. De aquí la necesidad de proveer servicios integrados que puedan ser combinados fácilmente con aplicaciones ubicadas en diferentes lugares geográficos. Los generadores de reportes en su mayoría posibilitan la comunicación e interoperabilidad con terceros, independientemente de sus propiedades o de las plataformas sobre las que se instalen. En este proyecto, la línea de reportes es la encargada de la generación de informes. Debido a la existencia de deficiencias se presenta la siguiente **situación problemática**:

El sistema de generación de informes del SCADA-GALBA, implementado sobre la biblioteca gráfica GTK, es un sistema que consume gran cantidad de recursos por lo que es muy lento, además no es capaz de asimilar nuevas funcionalidades como la portabilidad, dificulta el soporte, la corrección de errores, así como la realización de mejoras en el diseño del mismo y no puede ser accesible desde un navegador web.

De todo lo expuesto anteriormente se presenta como **problema científico** la siguiente interrogante:

¿Cómo visualizar los reportes procesados por el generador de informes del proyecto SCADA-GALBA en la web?

Para lo cual el **objeto de estudio** lo constituye: la visualización de informes.

El **objetivo general** del trabajo presentado en este informe es implementar un módulo de visualización web para el generador de informes del proyecto SCADA-GALBA.

Derivándose que el **campo de acción** de esta investigación lo constituye: la visualización de informes en la web.

Como **objetivos específicos** se identificaron:

- Desarrollar las funcionalidades necesarias para la visualización web del generador de informes del SCADA-GALBA.

- Desarrollar pruebas de unidad al módulo de visualización web del generador de informes del proyecto SCADA-GALBA.

En consecuencia, la **idea a defender** con este trabajo será: El proyecto SCADA-GALBA, al contar con un módulo de visualización web, podrá acceder a las funcionalidades del Generador de Informes desde un navegador.

Para ello se propusieron las siguientes **tareas**:

- Estudio y análisis de las herramientas y tecnologías utilizadas para conceptualizar el estado del arte del objeto de estudio.
- Investigar sobre las diferentes bibliotecas y tecnologías, con el fin de seleccionar las más adecuadas para el desarrollo de aplicaciones sobre plataformas libres.
- Estudiar y revisar los componentes del generador de informes del SCADA-GALBA para definir las funcionalidades a desarrollar.
- Implementar los elementos del sistema de visualización web del generador de informes que sean necesarios para la estructura de los controles y las vistas de la interacción del usuario con el sistema.
- Validar la propuesta del sistema mediante pruebas unitarias para verificar el correcto funcionamiento de la aplicación.

Durante el desarrollo de esta investigación se hizo necesario profundizar en el estudio de los temas abordados, por lo que se usaron varios **métodos teóricos** y **métodos empíricos** los cuales se describen a continuación:

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

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

Método histórico-lógico: Para la comprensión de los antecedentes y las tendencias actuales referidas a la evolución en el mundo de los sistemas SCADA Web.

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

Experimento: Elaboración de un módulo de visualización web para el generador de informes del proyecto SCADA.

Criterio de expertos: Permite la obtención del conocimiento necesario para saber lo referente al tema y continuar con la investigación, mediante encuestas o entrevistas, cuya experiencia y opiniones pueden ser de una valiosa contribución para el desarrollo del trabajo.

Los resultados esperados al culminar este trabajo de diploma son los siguientes:

- Obtener un módulo de visualización web para el generador de informes del proyecto SCADA-GALBA que cumpla con la calidad y las funcionalidades necesarias para su correcto funcionamiento.
- Con el diseño y desarrollo de las pruebas se espera un módulo que cumpla con los objetivos para los cuales fue desarrollado, además de un código con la calidad requerida.

El siguiente trabajo de diploma está estructurado de la siguiente forma:

Capítulo 1: En este capítulo se explican las principales características de los SCADA, haciendo énfasis en la generación de informes. Se realiza un estudio de las tendencias y tecnologías actuales a utilizar según las características del sistema.

Capítulo 2: Se analizan la arquitectura propuesta y se explican las principales funcionalidades.

Capítulo 3: Este capítulo se dedica a las pruebas de unidad realizadas para garantizar la integridad de la aplicación desarrollada.

Capítulo 1 **Fundamentación teórica.**

Introducción

En el presente capítulo se hace alusión a las características generales de los sistemas SCADA y su estructura. Además, se describen las principales funciones del SCADA, y los módulos que lo componen. Se hace referencia a la generación de informes en los SCADA y las generalidades sobre los generadores. Se realiza un análisis de las herramientas y tecnologías utilizadas en el desarrollo y documentación de la aplicación, así como las bibliotecas y lenguajes de programación estudiados en función de la elaboración del módulo de visualización web para el generador de informes.

1.1 Introducción a los sistemas SCADA-GALBA

El término SCADA usualmente se refiere a un sistema central que monitorea y controla un sitio completo o un sistema que se extiende sobre una gran distancia (kilómetros / millas). La instalación de un sistema SCADA necesita un hardware de señal de entrada y salida, sensores y actuadores, controladores, HMI (Interfaz Hombre Máquina), redes, comunicaciones, bases de datos, entre otros. Los sistemas SCADA mejoran la eficacia del proceso de monitoreo y control proporcionando la información oportuna para poder tomar decisiones operacionales apropiadas. De igual forma, como cuenta con información (alarmas, históricos, paradas, etc.) de primera mano de lo que ocurre u ocurrió en el proceso, permite la integración con otras herramientas del negocio como lo son intranets, ERP, entre otras.(1)

Un SCADA controla usualmente un sitio con gran extensión territorial. En realidad la mayor parte del control del sitio se realiza automáticamente por equipos electrónicos especializados para estos fines. Las funciones de control del servidor están casi siempre restringidas a reajustes básicos del sitio o a verificaciones de la operacionalidad de nivel de supervisión. El flujo de la información en los sistemas SCADA es como se describe a continuación: el fenómeno físico lo constituye la variable que se desea medir. La naturaleza de estas variables, dependiendo del proceso, la naturaleza del fenómeno y el modelo seleccionado para su supervisión, puede ser muy diversa: presión, temperatura, flujo de potencia, intensidad de corriente, voltaje, entre otros. Estos parámetros deben traducirse a una variable que sea entendible para el sistema SCADA, es decir, en una variable eléctrica. Una vez acondicionada la señal, la misma se convierte en un valor digital equivalente en el bloque de conversión de datos. Generalmente, esta función es llevada a cabo por un circuito de conversión analógico/digital. La computadora almacena

Capítulo 1: Fundamentación teórica

esta información, la cual es utilizada para su análisis y para la toma de decisiones. Simultáneamente, se muestra la información al usuario del sistema, en tiempo real. Basado en la información, el operador puede tomar la decisión de realizar una acción de control sobre el proceso. El operador comanda al computador a realizarla y de nuevo debe convertirse la información digital a una señal eléctrica.

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

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

A continuación se presenta un Esquema de un sistema de Adquisición, supervisión y control de datos.

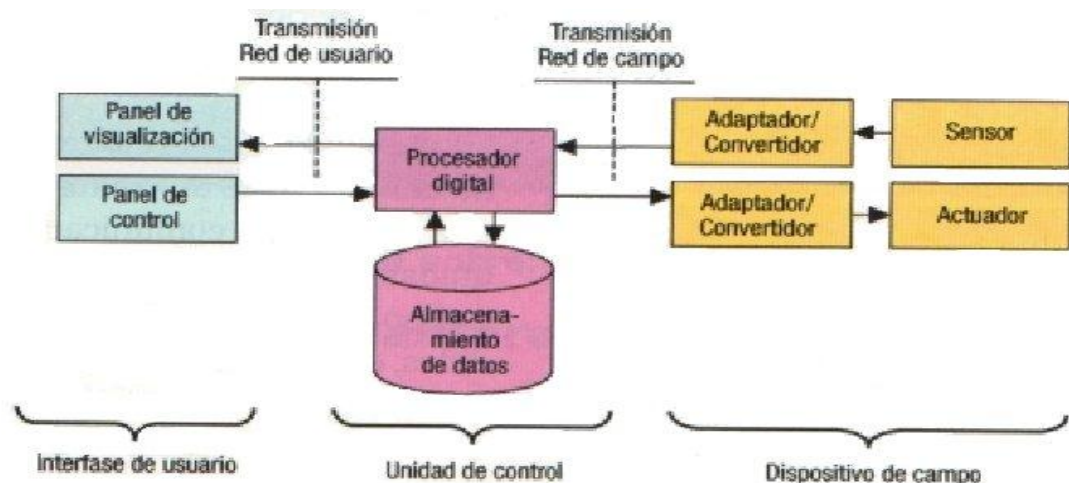


Fig. 1. Esquema de un Sistema de Adquisición, Supervisión y Control de Datos.

1.2 Módulos del SCADA-GALBA

➤ Base de Datos Históricas

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

➤ Middleware

Este módulo representa la capa de software encargada de la comunicación entre los diferentes procesos distribuidos, de mediano y alto nivel.

➤ Seguridad

Provee las interfaces necesarias para que los usuarios puedan autenticarse en el sistema, y de esta forma cada uno pueda acceder sólo a los recursos que tiene asignado su rol.

Además, posee herramientas para la protección ante ataques piratas, fallos eléctricos, problemas de red, servidores, entre otros.

➤ Reportes

En los SCADA existe la necesidad de contar con información sobre el comportamiento de variables en los distintos procesos a través del tiempo, este módulo es el encargado de brindar esta información al usuario a través de reportes especializados, los cuales se diseñan, teniendo en cuenta la necesidad de los usuarios, gracias a un editor que posee el mismo.

➤ Ambiente de Configuración

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

➤ Visualización (HMI)

Este módulo es el encargado de permitir representar gráficamente, de manera fidedigna los procesos operacionales con los que los usuarios interactúan. A través del mismo se pueden visualizar condiciones

operacionales, valores de variables, visualizar alarmas, navegar entre despliegues, enviar comandos, entre otros.

- Núcleo de procesamiento de datos (Recolector)

El núcleo de procesamiento de datos es el encargado del tratamiento y análisis en tiempo real de la información adquirida desde el campo.

- Manejadores

El módulo de manejadores de dispositivos asegura la comunicación del SCADA con los dispositivos de campo. Esta comunicación se realiza a través de una interfaz genérica única para todos los protocolos y bibliotecas dinámicas (Manejadores o drivers) que implementan esta interfaz en cada caso específico.

- Configuración

Este módulo se encarga de almacenar, persistir y suministrar la información base para el funcionamiento de los demás módulos del SCADA, (BDH, BDTR, HMI entre otros), cada módulo del sistema posee un agente (biblioteca), que permite establecer las comunicaciones con el servidor de configuración, a través de la capa de conectividad, permitiendo crear, eliminar y modificar los recursos configurables del SCADA.

1.3 Generación de informes en los sistemas SCADA

En la actualidad existe una gran cantidad de proyectos que han asumido como tarea el desarrollo de generadores de informes. Aunque ninguno de los encontrados se enfoca directamente en la generación de informes web para un SCADA. Un generador de informes en un sistema de este tipo es la herramienta que permite emitir informes en un formato personalizado de tal manera que le sea útil al personal al cual va dirigido, igual a cualquier sistema de gestión, aunque con algunas peculiaridades propias de esta aplicación. Este sirve de puente entre la base de datos histórica del SCADA y los diseños de informes que el usuario previamente configura de acuerdo a sus necesidades para volcar, filtrar y analizar datos.

De aquí que un estudio de los resultados alcanzados en estos proyectos tenga una gran utilidad para desarrollar el generador de informes web necesario para el generador de informes del proyecto SCADA, incluso, teniendo presente la posible adopción de algunos resultados de estos proyectos como soluciones para el que nos ocupa.

A pesar de la existencia de varios proyectos desplegados sobre software libre para el desarrollo de generadores de informes, no deben descartarse las ideas que pudieran tomarse de algún sistema propietario. (2)

1.3.1 Generalidades sobre los generadores de informes

De forma general un generador de informes se compone de dos elementos básicos, un diseñador de informes y un motor de generación. El funcionamiento del subsistema encargado de la generación de reportes puede dividirse en dos:

- Por un lado un diseñador de informes, que proporciona al usuario una forma sencilla de diseñar una plantilla, que se utilizará posteriormente en la generación del reporte.
- Por otro lado, existe un motor para la generación de reportes que se encarga de extraer los datos de una o varias bases de datos, después enlaza estos datos con el diseño realizado antes y finalmente genera un documento en base al esquema (diseño) del informe realizado con el diseñador. El documento resultante puede ser un archivo PDF u otro formato, que incluye todo lo citado anteriormente (diseño + datos), el cual se puede visualizar en pantalla, imprimir o simplemente ser almacenado en algún dispositivo de almacenamiento digital.

En la figura (Fig.2) se muestra de forma esquemática el funcionamiento del sistema en el cual se pueden apreciar los dos grandes componentes del subsistema a los que responde el funcionamiento anterior.

Como se puede inferir de esta división de las funcionalidades del subsistema en dos partes, el proceso de generación de reportes en el SCADA también estará estructurado por dos etapas o momentos:

- Etapa de diseño: En ésta el mantenedor crea los diseños que posteriormente serán usados por el motor de generación, para esto el subsistema proporciona una herramienta que permite la creación visual e interactiva de los diseños que posteriormente podrán ser usados para la generación de reportes.
- Etapa de generación: Este es el momento en el cual los usuarios generan el reporte deseado, tramando como fuente el diseño correspondiente y los valores almacenados en la base de datos.

Para acceder a cada una de las etapas será necesario contar con los privilegios como usuario que permitan hacer uso de las funcionalidades implicadas en cada momento, así, para usar las funcionalidades del diseñador, será necesario estar autenticado en el sistema con privilegios de mantenedor. Al igual se tendrá que tener acceso a los recursos involucrados en cada momento como es el caso del acceso a la base de datos y al diseño involucrado en el momento de generación del reporte. (3)

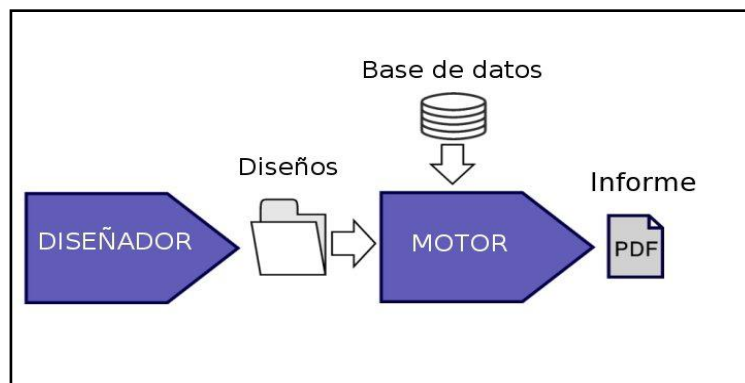


Fig. 2. Estructura general de un generador de informes.

De aquí se puede deducir que las principales tareas que debe realizar un generador de informes son las siguientes:

- Diseñar la apariencia que van a tener los informes.
- Extraer la información y volcarla de forma ordenada con la apariencia diseñada.

Haciendo uso del diseñador se obtienen plantillas de diseños que posteriormente son empleadas por el motor de generación para volcar información proveniente de almacenes de datos y obtener el informe final con la apariencia preestablecida en el diseño.

Existen algunos proyectos que solo desarrollan el motor de generación. En este caso también se considera la aplicación como un generador de informes, pero se debe tener bien presente la necesidad de brindar el diseño al motor en algún formato entendible por este, lo cual, de no contarse con un diseñador, puede resultar un proceso muy engorroso. De aquí que la presencia de un diseñador que brinde flexibilidades para diseñar los informes que posteriormente se generará, sea importante, sino imprescindible, en una aplicación como esta. (4)

1.4 Evaluación de generadores de informes web

La principal característica a tener en cuenta en cualquier proyecto, a la hora de reutilizar software desarrollado por otras personas, es la liberación del código que acompaña la aplicación. De aquí que el análisis que se presenta a continuación se divide en generadores de informes propietarios (no se puede tener acceso a su código) y generadores de informes desarrollados bajo los principios de software libre.

1.4.1 Generadores de informes propietarios web

1.4.1.1 Stimulsoft Reports.Web

Stimulsoft Reports.Web es una herramienta diseñada para crear, mostrar, imprimir y exportar informes a través de un navegador web, poniendo a su disposición el ciclo completo del diseño de informes sin que se deba cerrar el navegador utilizado. Es importante decir además, que es el primer generador que permite editar informes directamente en la web. No es necesario instalar el .Net Framework, componentes ActiveX o algún plugin en específico en la máquina cliente, todo lo que se necesita es un navegador web con Flash Player 9 instalado.

Stimulsoft Reports permite exportar los informes a más de 20 formatos: PDF, XPS, Excel, Word, HTML, RTF, XML entre otros. Todo esto se puede hacer desde un componente basado en Ajax, Web Viewer, y directamente desde código de la aplicación. No presenta ningún problema en obtener datos desde ficheros XML y es capaz de procesar los datos: ordenar, filtrar, agrupar, juntar y sumar.(5)

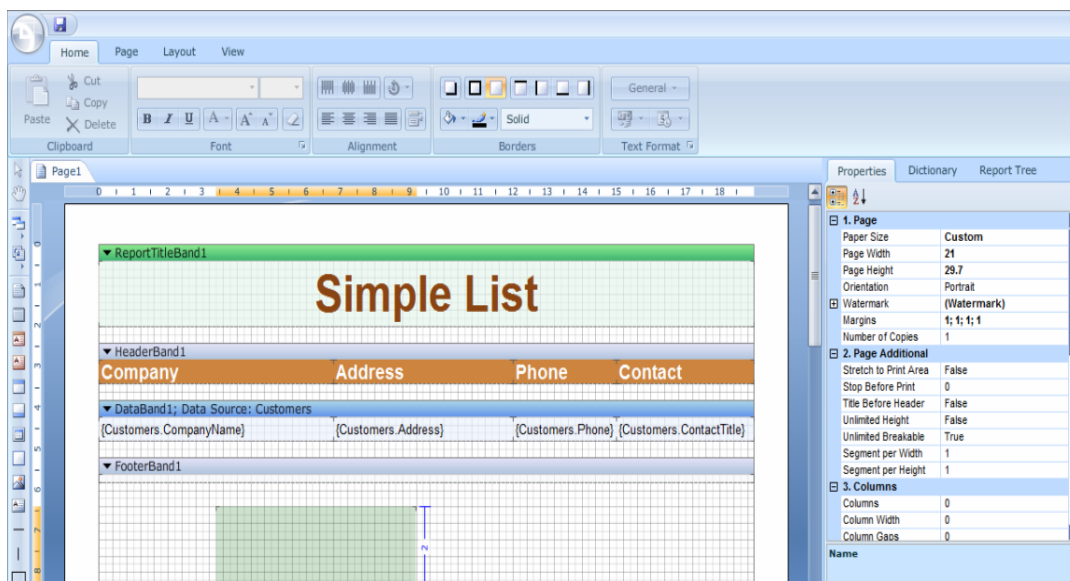


Fig.3. Report Designer in Web.

1.4.1.2 XML Report Generator (XMLRG).

XML Report Generator (XMLRG) es una aplicación web que utiliza datos de las empresas Oracle DB 9i/10g para fusionarlas en un informe y generarlos en formato PDF, PostScript o formatos de la AFP basado en plantillas ya preparadas en Microsoft Word 2003/2007, utilizando DataStyler. Cada informe puede contener cualquier número de plantillas de diseño, para la combinación de datos se puede registrar y luego se usa para la generación de informes. Brinda la posibilidad de la creación dinámica de los formularios introduciendo los datos descritos por el esquema XSD de los datos para insertar, entrada de datos por un usuario y la generación de un informe adecuado. La aplicación es compatible con funciones administrativas, como la creación y edición de la cuenta de usuario, acceso a los informes de derechos de usuario, la adición y modificación de las listas y los atributos de los informes de casos. La utilidad de instalación se maneja para establecer la parte del servidor del producto, que casi automatiza completamente la instalación de la aplicación. La parte del servidor de la aplicación se puede instalar en diferentes plataformas - Windows, Linux. XML Generador de Reportes (XMLRG) es JDBC compatibles con el apoyo a XML API. (6)

Year, Month	Total
200606	\$1,295,820.32
200609	\$1,307,195.25
200608	\$1,339,067.54
200611	\$1,270,868.73

Fig.4. XML Report Generator.

1.4.1.3 ReportAllWeb

Es una aplicación diseñada para generar reportes web desde la base de datos de un sistema de información (SQLServer, Oracle, Postgres, Access, ODBC, ADO). Permite conectarse a la base de datos de su sistema de información, obtener un conjunto de resultados para el reporte y visualizarlos en el

Capítulo 1: Fundamentación teórica

navegador, estableciendo criterios de búsqueda, ordenamiento, subtotales y ventanas para detalle de datos, entre otros.

Con esta aplicación se puede:

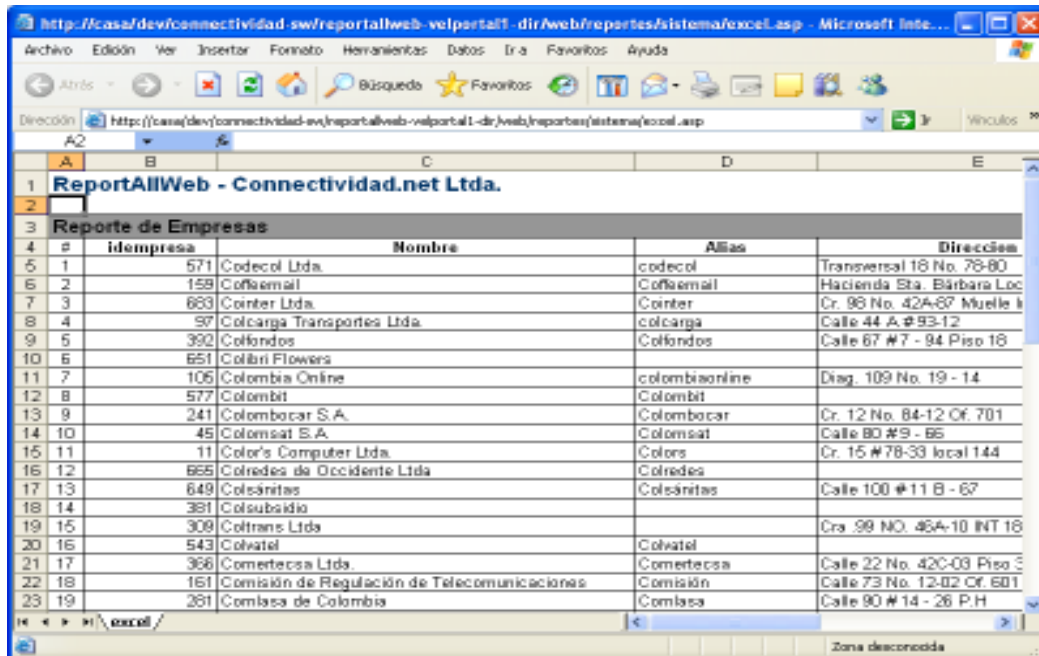
- Decidir que tablas y/o vistas utilizar de su base de datos de origen.
- Seleccionar las columnas a usar, a visualizar, el tipo de sus datos, las opciones para criterios de búsqueda y las sumas de columnas en subtotales.
- Definir reportes de detalle (Drill down/up), en ventanas emergentes a modo de tabla con múltiples entradas o a modo de lista de datos de una única entrada, y relacionarlos entre sí por medio de vínculos.
- Generar reportes que ofrezcan la posibilidad de navegar por un universo de información con significado.
- Agrupar los datos y generar subtotales por múltiples niveles.
- Realizar ordenamientos por cualquier columna.
- Exportar los resultados a Excel.
- Predefinir reportes y guardarlos como opciones directas del menú, para que sean utilizados recurrentemente sin necesidad de establecer cada vez los criterios de búsqueda y las opciones de visualización.
- Ofrecer acceso anónimo a los reportes predefinidos (acceso público sin autenticación previa).
- Opera con MS-MSQLServer 2000/2005, Oracle 9i/10g, Postgres 8.x, Access 2000, ODBC y en general cualquier fuente de datos con acceso vía ADO.
- Se integra con su aplicación Web de manera sencilla en la intranet o en Internet.

Da la posibilidad a cualquier usuario autenticado de generar reportes de acuerdo a sus necesidades, especificando los criterios de búsqueda para filtrar el resultado y estableciendo las opciones de ordenamiento y agrupación de columnas para la visualización de subtotales. Si la consulta seleccionada como fuente de datos del reporte tiene vínculos definidos (Drill down/up), el usuario podrá hacer uso de ellos para navegar a través de la información disponible.

#/Vínculos	idempresa	Nombre	Alias	Direccion	
1	571	Codecol Ltda.	codecol	Transversal 19 No. 79-60	219221
2	159	Coffeeemail	Coffeeemail	Hacienda 55a. Barrabá Local A-122	61207-
3	603	Cointar Ltda.	Cointar	Cr. 99 No. 42A-87 Nueva Ind. Bodega 41	41059-
4	97	Colcarga Transportes Ltda.	colcarga	Calle 44 A # 93-12	415301
5	392	Colfondos	Colfondos	Calle 67 # 7 - 94 Piso 19	
6	651	Colibri Flores			842690
7	105	Colombia Online	colombiaonline	Diap. 109 No. 19 - 14	657949

Fig. 5. Ejemplo de un reporte generado que tiene vínculos definidos (Drill down/up).

Permite exportar a Excel cualquier reporte generado que se visualice en el navegador, para graficar los resultados o realizar operaciones propias de la hoja de cálculo. (7)



The screenshot shows a web browser window with the address bar containing the URL: <http://casa/dev/connectividad-sw/reportallweb-vportal1-dir/web/reportes/sistema/excel.asp>. The browser's menu bar includes 'Archivo', 'Edición', 'Ver', 'Insertar', 'Formato', 'Herramientas', 'Datos', 'Ir a', 'Favoritos', and 'Ayuda'. The spreadsheet content is as follows:

Reporte de Empresas				
#	idempresa	Nombre	Alias	Dirección
1	571	Codecol Ltda.	codecol	Transversal 16 No. 78-80
2	159	Colfaemail	Colfaemail	Hacienda Sta. Bárbara Loc
3	683	Cominter Ltda.	Cominter	Cr. 96 No. 42A-67 Muelle 1
4	97	Colcarga Transportes Ltda.	colcarga	Calle 44 A #93-12
5	392	Colfondos	Colfondos	Calle 67 #7 - 94 Piso 16
6	651	Colibri Flowers		
7	106	Colombia Online	colombiaonline	Diag. 109 No. 19 - 14
8	577	Colombit	Colombit	
9	241	Colombocar S.A.	Colombocar	Cr. 12 No. 84-12 Of. 701
10	45	Colomsat S.A.	Colomsat	Calle 80 #9 - 66
11	11	Color's Computer Ltda.	Color's	Cr. 15 #78-33 local 144
12	666	Colredes de Occidente Ltda	Colredes	
13	649	Colánitas	Colánitas	Calle 100 #11 B - 67
14	381	Colsubsidio		
15	309	Coltrans Ltda		Cra. 99 NO. 46A-10 INT 18
16	543	Colviatel	Colviatel	
17	366	Comertecsa Ltda.	Comertecsa	Calle 22 No. 42C-03 Piso 3
18	161	Comisión de Regulación de Telecomunicaciones	Comisión	Calle 73 No. 13-02 Cf. 601
19	261	Comlisa de Colombia	Comlisa	Calle 90 #14 - 26 P.H

Fig. 6. Reporte exportado a Excel.

1.4.2 Generadores de informes bajo los principios del Software Libre

1.4.2.1 JasperReport

JasperReports es una aplicación web que presenta una flexible solución de código abierto para la generación y gestión de informes. Consta de un módulo que dispone de un depósito de archivos que usa un sistema de carpetas, una aplicación que muestra todos los informes que están en el depósito y un visor de dichos informes, además permite subir todos los informes creados e inmediatamente estos están disponibles para todos los usuarios.

El principal objetivo de esta herramienta es facilitar la construcción de documentos con contenido dinámico y su visualización en diferentes formatos (PDF, HTML y XML). También existe la posibilidad de exportar los informes a una hoja de cálculo de Excel o a un documento Word.

JasperReports no requiere grandes requisitos para ponerle en marcha, la única configuración que es necesario instalar es la que conecte el módulo con la base de datos. (8)

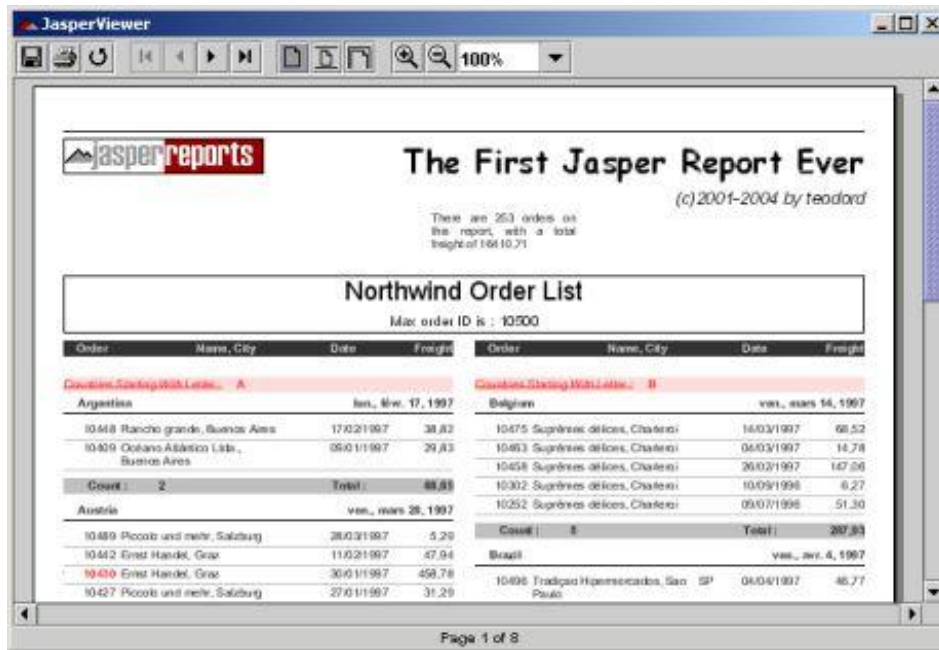


Fig. 7. Representación de la interfaz de la Aplicación Jasper Report Ever.

1.5 Tendencias y tecnologías actuales a considerar

1.5.1 Software Libre

El software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software; de modo más preciso. Ofrece cuatro libertades a los usuarios del software: la libertad de usar el programa, con cualquier propósito; de estudiar su funcionamiento, y adaptarlo a las necesidades; de distribuir copias, con lo que puede ayudar a otros; de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al código fuente es un requisito previo). El autor del software concede a cualquiera el derecho a usar su obra, a modificarla y a adaptarla a sus necesidades específicas. Este derecho de libre uso a veces se otorga sin restricciones (licencias tipo BSD) y otras veces con la única condición de que toda mejora se distribuya con las mismas condiciones de uso que tiene originalmente y por tanto siga estando libremente disponible para todo el mundo (licencias tipo copyleft, como la GPL de la Free Software Foundation).(3)

1.5.2 Ambiente de Desarrollo Integrado (IDE)

El ambiente de desarrollo integrado es un elemento imprescindible en la producción de software. Es donde se definen el conjunto de herramientas y tecnologías (frameworks), versiones a usar y su integración, que intervienen en un proceso de desarrollo de software. Un entorno de desarrollo integrado o Integrated Development Environment (IDE, por sus siglas en inglés), es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word. Los IDEs proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Java, C#, Delphi, Visual Basic y Object Pascal. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación, este es el caso de Eclipse, que mediante pluggins se le pueda añadir soporte de lenguajes adicionales. Otra característica importante para un IDE es que puede funcionar como un sistema en tiempo de ejecución en algunos lenguajes de programación. (9)

1.5.2.1 Eclipse

Es uno de los entornos integrados de desarrollo (IDE), de código abierto y extensible, más utilizado en los últimos tiempos. Este incluye varias características únicas, como la refactorización de código, la actualización/instalación automática de código (mediante Update Manager), una lista de tareas, soporte para unidades de test con JUnit, e integración con la herramienta de construcción de Jakarta: Ant.

El entorno integrado de desarrollo (IDE) de Eclipse emplea módulos (en inglés *plugin*) para proporcionar toda su funcionalidad. Este mecanismo de módulos es una plataforma ligera para componentes de software. Por otra parte, permite trabajar con lenguajes para procesamiento de texto como LaTeX, aplicaciones en red como Telnet y sistema de gestión de base de datos. La arquitectura plugin permite escribir cualquier extensión deseada en el ambiente, como la gestión de la configuración. Se provee soporte para Java y el sistema de control de versiones en el SDK de Eclipse. (3)

Ventajas:

- Eclipse es adaptable a cualquier tipo de lenguaje, por ejemplo C/C++, Cobol, C#, XML, PHP, entre otros.
- La característica clave de Eclipse es la extensibilidad.
- Compilación en tiempo real.
- La depuración e implementación de aplicaciones resultan mucho más sencillas.
- Es multiplataforma.

1.5.2.2 Code::Blocks

Code::Blocks es un entorno de desarrollo integrado libre y multiplataforma para el desarrollo de programas en lenguaje C++. Está basado en la plataforma de interfaces gráficas WxWidgets, lo cual quiere decir que puede usarse libremente en diversos sistemas operativos, y está licenciado bajo la Licencia pública general de GNU. Este IDE fue construido como un núcleo altamente expansible mediante complementos (pluggins). Actualmente la mayor parte de la funcionalidad viene provista por los complementos incluidos predeterminadamente. No es un IDE autónomo que acepta complementos, sino que es un núcleo abstracto donde los complementos se convierten en una parte vital del sistema. Esto lo convierte en una plataforma muy dinámica y potente, no solo por la facilidad con que puede incluirse una nueva funcionalidad, sino por la capacidad de poder usarla para construir otras herramientas de desarrollo tan solo añadiendo complementos.(10)

1.5.3 Cliente Web

En la arquitectura cliente/servidor conocida como una arquitectura de red que separa el cliente del servidor, se denomina cliente(a menudo una aplicación que usa interfaz gráfica) al proceso que inicia el diálogo o solicita los recursos. Específicamente un cliente web es una aplicación que funciona sobre un navegador estándar, recibe y muestra los datos del servidor. Mediante la aplicación cliente los usuarios interactúan con el proceso. La comunicación con el servidor web se realiza a través del protocolo HTTP, el cual permite la transferencia de archivos hipertexto.

En general, una aplicación que inicia una comunicación con otra se denomina cliente. Los usuarios finales invocan aplicaciones cliente cuando utilizan un servicio de red. Cada vez que se ejecuta una aplicación cliente, esta contacta con el servidor, le envía una solicitud de servicio y espera la respuesta o resultados del servicio. Este proceso es el encargado de llevar a cabo la interacción con el usuario y de mostrar los

resultados de las peticiones de servicio. En la mayoría de las ocasiones los clientes son más fáciles de diseñar que los servidores, y no suelen precisar privilegios especiales del sistema para poder funcionar.

1.5.4 Servidor Web

Un servidor web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada). Un servidor web básico cuenta con un esquema de funcionamiento muy simple, basado en ejecutar infinitamente el siguiente bucle:

- Espera peticiones en el puerto TCP indicado (el estándar por defecto para HTTP es el 80).
- Recibe una petición.
- Busca el recurso.
- Envía el recurso utilizando la misma conexión por la que recibió petición.
- Vuelve al segundo punto.

Un servidor web que siga el esquema anterior cumplirá todos los requisitos básicos de los servidores HTTP, aunque sólo podrá servir ficheros estáticos.

A partir del anterior esquema se han diseñado y desarrollado todos los servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGIs, Servlets, entre otros) que pueden atender, en función de que sean o no sean multi-proceso.(2)

1.5.5 Protocolo de comunicación HTTP

El Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol*) es un protocolo cliente-servidor que manipula los intercambios de información entre los clientes web y los servidores HTTP. Este es orientado a transacciones y sigue el esquema de petición-respuesta entre un cliente y un servidor. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Una vez que se establece la conexión, el protocolo TCP (*Transmission Control Protocol*) se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

Una transacción HTTP está formada por un encabezado seguido, opcionalmente, por el dato. El encabezado especificará la acción requerida del servidor, el tipo de dato retornado, o el código de estado. El uso de campos de encabezados enviados en las transacciones, le da gran flexibilidad al protocolo HTTP y permitiendo la autenticación, cifrado e identificación de usuario.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores ya que no recuerda el estado entre una solicitud y otra siguiente, cada solicitud es independiente a otra. El desarrollo de aplicaciones web necesita frecuentemente mantener estado, para esto se usan las cookies, que es información que un servidor puede almacenar en el sistema cliente. Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.(11)

1.5.6 Lenguajes de programación para la web

Entre los distintos lenguajes de programación para la web que existen en la actualidad, se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente / Servidor. El primer grupo está formado por los lenguajes que se ejecutan en el servidor (en inglés, server side languages). Como ejemplos más sobresalientes tenemos algunos como PERL, ASP, PHP, Java, JSP, los módulos CGI, entre otros. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a bases de datos, tratamiento de la información, y demás. Dentro del segundo grupo se encuentran aquellos lenguajes que se ejecutan en el cliente (client side languages). En este caso están el JavaScript y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores.

Es importante seleccionar el lenguaje a utilizar, tanto del lado del servidor como del lado del cliente. A continuación se hace un breve análisis de los lenguajes de uso más común en la actualidad, con vistas a hacer una apropiada selección.

1.5.6.1 Perl (Practical Extracting and Reporting Language)

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para la Web, y es considerado el lenguaje perfecto para este fin dadas sus facilidades en cuanto a la manipulación de texto. Es un lenguaje de libre uso. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP.(12)

1.5.6.2 Lenguaje HTML

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español

Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html). Es sencillo y permite describir hipertexto. El texto es presentado de forma estructurada y agradable. No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web. Es un lenguaje de fácil aprendizaje y lo admiten todos los exploradores. Presenta desventajas por ser un lenguaje estático, porque el diseño es lento y las etiquetas son muy limitadas.(12)

1.6 Framework web

En el desarrollo de software, un framework web se puede definir como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas web. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje de scripting para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. La mayoría de los frameworks web se encargan de ofrecer una capa de controladores de acuerdo al patrón MVC o con el modelo 2 de servlets, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación. (9) ([Anexo 3](#))

1.6.1 C++ Web Toolkit

Wt (pronunciado 'witty') es una librería C++ para el desarrollo de aplicaciones web interactivas.

La API es widget-céntrica e inspirada por escritorio GUI API. Ofrece al desarrollador completa abstracción de cualquiera de los detalles específicos de la implementación web, incluyendo el manejo de eventos, gráficos de apoyo, y las URL.

A diferencia de muchas páginas basadas en frameworks, Wt fue diseñado para crear aplicaciones con estado que son al mismo tiempo, altamente interactivas (aprovechando técnicas como AJAX al máximo) y accesible (con soporte para los navegadores HTML normal). Viene con un servidor de aplicación que actúa como un servidor independiente que se integra a través de Internet o FastCGI con otros servidores web.

Una aplicación web desarrollada con Wt está escrita en un solo lenguaje compilado (C++), de la que la biblioteca genera el necesario HTML / XHTML, JavaScript, CGI, SVG / VML / Canvas y el código AJAX. La

Capítulo 1: Fundamentación teórica

responsabilidad de la escritura segura y el navegador portátil de aplicaciones web es manejada por Wt. Por ejemplo, si está disponible, Wt utiliza JavaScript y AJAX al máximo, pero las aplicaciones desarrolladas con Wt también funcionarán correctamente cuando AJAX no está disponible, o JavaScript está desactivado, además mejora de manera progresiva de un HTML plano / aplicación CGI para una aplicación AJAX rica.(13)

Ventajas:

- Desarrolla aplicaciones web, como aplicaciones de escritorio.
- Proporciona un amplio conjunto de widgets, que funcionan independientemente de la disponibilidad de JavaScript (pero se benefician de la disponibilidad de JavaScript).
- Una especificación única para la validación y el manejo de eventos tanto para el cliente como para el servidor.
- Genera cumplimiento de normas o código XHTML.
- Portátil, suavizan gráficos (utiliza VML en línea, SVG en línea o el Canvas HTML 5).
- Evita los problemas de seguridad común, ya que tiene control completo sobre la capa de presentación, de los filtros de las etiquetas activas y los atributos de la exposición lógica de negocio que permanecen en el servidor.
- Tiempo de carga rápido y bajo uso del ancho de banda, que sólo se ven afectados por la complejidad de la pantalla, no por el tamaño de la aplicación.
- Una API simple para el servidor de eventos.
- Utiliza el built-in httpd para un fácil desarrollo e implementación, o usa el conector FastCGI para implementar una aplicación FastCGI en los servidores web existentes.

Características:

- Soporta todos los principales navegadores (Firefox / Gecko, Internet Explorer, Safari, Chrome, Konqueror y Opera), pero también navegadores planos de HTML (probado con Links y Lynx).
- Desarrollo e implementación en Unix / Linux GNU (gcc) o Microsoft Windows (Visual Studio).
- Igualdad de conducta, con o sin soporte para JavaScript o AJAX, en la medida de lo posible, mediante la degradación de gracia o la mejora progresiva (este último mediante el método de arranque progresivo).
- Soporte integrado para Unicode (UTF8/UTF16).

- Cuando AJAX está disponible, se representan sólo los cambios.
- Dos fases de representación de los cambios en la página web, en primer lugar, visible, y en el fondo los cambios invisibles (cuando AJAX está disponible). Cuando el AJAX no está disponible, se presenta sólo el contenido visible.
- Período de sesiones de seguimiento de opciones configurables, que incluyen la reescritura de URL y las cookies.
- Apoyo para la navegación por el historial del navegador (atrás / adelante botones y marcadores), bastantes URL, y la optimización de motores de búsqueda con un comportamiento unificado para HTML plano o sesiones de AJAX.

1.7 Lenguajes de programación

1.7.1 C++

C++ es un lenguaje de programación, diseñado a mediados de los años 1980, por Bjarne Stroustrup, como extensión del lenguaje de programación C. Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Existen también algunos intérpretes como ROOT.

Las principales características del C++ son el soporte para programación orientada a objetos, el soporte de plantillas o programación genérica (*templates*), la portabilidad, brevedad, programación modular, compatibilidad con C y velocidad. Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos. Además, se trata de un lenguaje de programación estandarizado (ISO/IEC 14882:1998), ampliamente difundido, y con una biblioteca estándar C++, que lo ha convertido en un lenguaje universal, de propósito general, y ampliamente utilizado tanto en el ámbito profesional como en el educativo.

Además, posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel entre las cuales se destacan la posibilidad de redefinir los operadores (sobrecarga de operadores) y la identificación de tipos en tiempo de ejecución (*RTTI*). C++ está considerado por muchos como el lenguaje más potente, debido a que permite trabajar tanto a alto como a bajo nivel, sin embargo, es a su vez uno de los que trae menos automatismos (obliga a hacerlo casi todo manualmente al igual que C) lo que dificulta mucho su aprendizaje.(3)

1.8 Biblioteca CxxTest.

Es una biblioteca utilizada para llevar a cabo las pruebas al código fuente de cualquier proyecto. Está diseñada para ser tan portátil como sea posible. Es muy fácil de usar y de poner en funcionamiento.

CxxTest utiliza el lenguaje Perl para generar código C++. Analiza el código de las pruebas implementadas por el usuario y genera un corredor de prueba en C++ que se ejecuta directamente desde las pruebas del usuario. Por su flexibilidad es posible hacer pruebas rigurosas con código sencillo. Su único requerimiento es instalar Perl y que este se ejecute correctamente.

Dentro de sus principales ventajas están:

- No requiere de funciones miembro de plantilla.
- No requiere de manejo de excepciones.
- No requiere ninguna biblioteca externa.
- Es una herramienta multiplataforma y está disponible en los repositorios de varias distribuciones de software libre como es el caso de Debian. Se distribuye en su totalidad como un conjunto de archivos de cabecera que le hace extremadamente portátil y utilizable.(14)

1.9 Biblioteca Pqxx

Biblioteca que posibilita la comunicación de aplicaciones que utilizan C++ como lenguaje de programación con el motor de base de datos PostgreSQL. El motor de la base de datos puede ser local o estar en otro equipo, accedido a través de TCP/IP.

1.10 Herramientas CASE

Las herramientas CASE (siglas en inglés de Computer Aided Software Engineering) son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores durante todos los pasos del ciclo de vida del software. Constituyen un soporte automatizado para el desarrollo y mantenimiento del software. Se pueden ver como la unión de las herramientas automáticas de software y las metodologías de desarrollo de software formales.

Existen múltiples herramientas de este tipo especializadas con diferentes propósitos: en la fase de diseño, para generar código, algunas tienen una visión de desarrollo orientada a procesos sin la capacidad de modelamiento, mientras otras proveen el modelamiento sin incluir los procesos de Análisis o Diseño. Entre

las herramientas CASE más conocidas y utilizadas a escala internacional, podemos mencionar ERWIN, EasyCASE, OracleDesigner, PowerDesigner, Rational Rose y Visual Paradigm. A continuación se analizarán las más utilizadas en la UCI. (16)

1.10.1 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Los sistemas de modelado UML ayudan a una más rápida construcción de aplicaciones de calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código a partir de diagramas, generación de objetos a partir de bases de datos, generación de bases de datos a partir de diagramas de entidad relación y generar documentación. Permite interoperabilidad con modelos UML2 (metamodelos UML 2.x para plataforma Eclipse). Se integra con Visio. Posee una plataforma, llamada SDE, capaz de integrarse con Eclipse, NetBeans, Oracle JDeveloper, JBuilder, IntelliJ IDEA, WebLogic Workshop, Microsoft Visual Studio, entre otras. (18)

1.10.2 IBM Rational Software Architect (RSA)

RSA es una herramienta colaborativa que provee de un soporte integral para el diseño y el desarrollo enfocado en un desarrollo dirigido por modelos con el lenguaje UML. Permite realizar un diseño visual de los sistemas a desarrollar, realizar transformaciones desde los diagramas a código y continuar con el desarrollo de la aplicación. Cubre el ciclo de vida del desarrollo basándose en la metodología *Rational Unified Process*.

Permite transformaciones de modelo a modelo y de modelo a texto, abarcando diferentes lenguajes de programación. Permite también la creación de patrones que reducen el esfuerzo de rehacer trabajos repetitivos, reduciendo costes de desarrollo y aumentando la calidad del software resultante. Soporta algunos de los estándares definidos por OMG, tales como UML, XMI y OCL.

Tiene integrado un entorno de desarrollo Eclipse que permite refinar el código generado de forma cómoda. De hecho, los productos de RSA están contruidos sobre una arquitectura extendida proporcionada por Eclipse, por lo que su integración es total a través de un conjunto de pluggins.

Dentro de los componentes Eclipse integrados en RSA, el que más estrechamente relacionado está con el modelado es EMF, marco de trabajo que permite generar código eficiente y adaptable desde los propios modelos. También integra *Eclipse Modeling Framework Technologies* (EMFT) que contempla diferentes tecnologías que complementan la manipulación de modelos basados en EMF y otros subproyectos de Eclipse Modeling Project como GMF y GEF.(19)

1.11 Metodologías de Desarrollo de Software

El uso de las metodologías de desarrollo de software persigue como objetivo principal, lograr el éxito rotundo de los proyectos de producción de software, para lo cual imponen un proceso disciplinado con el fin de hacerlo más predecible y eficiente.

Hasta hace poco el proceso de desarrollo llevaba asociado un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Este esquema tradicional para abordar el desarrollo de software ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto a tiempo y recursos), donde por lo general se exige un alto grado de ceremonia en el proceso. Sin embargo, este enfoque no resulta ser el más adecuado para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del “buen hacer” de la ingeniería del software, asumiendo el riesgo que ello conlleva.

A continuación se analizarán varias de las metodologías de desarrollo de software más utilizadas en el mundo, ya sean ágiles, pesadas o tradicionales, definiendo sus principales características, así como localizando las principales ventajas y desventajas que presentan.

1.11.1 RUP

El proceso unificado (Unified Process o UP) es un marco de desarrollo de software iterativo e incremental. A menudo, es considerado como un proceso altamente ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto de software. Dado que es un marco de procesos, puede ser adaptado y el más conocido es RUP (Rational Unified Process) de IBM.

El Proceso Unificado de Rational (Rational Unified Process), es un proceso de desarrollo de software que engloba elementos de todos los fundamentos de proceso comentados dentro del apartado de metodología, además de una serie de buenas prácticas que han de observarse para la obtención de un producto de calidad. Dependiendo de las características de cada proyecto, se adapta, relajando aquellas partes que lo pueden hacer excesivamente rígido o burocrático. En RUP se puede ver la evolución del software en cuatro fases, al final de las cuales, y tras una serie de iteraciones, establece objetivos a alcanzar bien definidos, cada una de estas fases se completa con la realización de varias iteraciones en las que se desarrollan una serie de actividades, que el modelo RUP clasifica en 9 disciplinas que tienen más o menos importancia en función de lo cerca que se esté o no de la finalización del proyecto. Establece además lo que denomina buenas prácticas, como forma de trabajo adecuada para la obtención de objetivos que se pueden ir perfeccionando.

Finalmente, cabe indicar que el modelo establece que el propio proceso es adaptable a cada caso de desarrollo. Por ejemplo, no todos los proyectos requieren del mismo nivel de documentación. El tamaño, la complejidad y el número de participantes entre otros aconsejan definir para el proyecto cómo adaptar el proceso. En la adaptación se definen qué artefactos hay que producir y en qué detalle, qué roles intervienen y las funciones que desempeñan dentro del proyecto.(20)

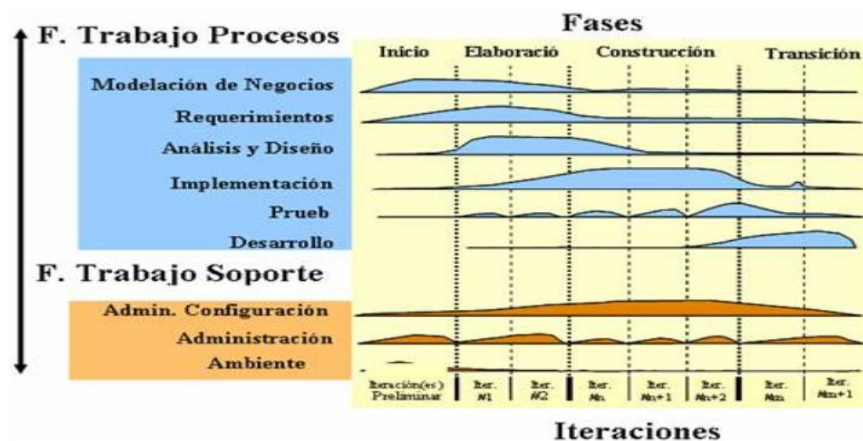


Fig.8. Fases de RUP.

1.12 Propuesta de solución

Luego del análisis llevado a cabo, se puede plantear una propuesta que consiste en desarrollar un módulo de visualización web del generador de informes del proyecto SCADA mediante una alternativa autónoma,

Capítulo 1: Fundamentación teórica

que siga el modelo de los generadores de informes existentes y que incluya además otras funcionalidades necesarias, utilizará como IDE para desarrollo Eclipse por ser de código abierto, extensible y porque incorpora funcionalidades como la refactorización de código, para la interfaz visual se decidió utilizar el framework C++ Web Toolkit el cual ya tiene incorporado a Extjs que es una librería JavaScript ligera y de alto rendimiento, compatible con la mayoría de los navegadores que sirve para crear páginas e interfaces web dinámicas, utilizará como lenguaje de programación a C++ dada su portabilidad y eficiencia. Se decidió la utilización de la metodología RUP porque provee un entorno de proceso de desarrollo configurable, basado en estándares; permite tener claro y accesible el proceso de desarrollo que se sigue; puede ser configurado a la necesidades de la organización y del proyecto, para el modelado se utilizará Visual Paradigm esta es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. La aplicación será desarrollada sobre software libre por las ventajas que trae esto consigo de reutilización de código, independencia tecnológica entre otras.

1.13 Conclusiones

En este capítulo se hizo alusión a las características generales de los sistemas SCADA y su estructura. Además, se describieron las principales funciones del SCADA-GALBA y los módulos que lo componen. Se hizo referencia a la generación de informes en el SCADA-GALBA y a las generalidades sobre los generadores. Se realizó un análisis de las tecnologías a utilizar en el desarrollo de la propuesta de solución. Se fundamentó la elección del lenguaje de programación, las herramientas, el entorno de desarrollo y la metodología de desarrollo de software y finalmente, se planteó la propuesta que incluye dichos aspectos. A partir de este capítulo, se abordará una descripción de la propuesta de solución.

Capítulo 2: Descripción y construcción de la solución propuesta.

Capítulo 2 **Descripción y construcción de la solución propuesta.**

Introducción

En este capítulo se hace una descripción de las principales características que corresponden a la implementación del módulo de visualización web del generador de informes del proyecto SCADA-GALBA. En el primer epígrafe se muestra la arquitectura y los patrones que fueron utilizados. Posteriormente se hace una descripción de las principales funcionalidades que se implementaron en la construcción del software así como los principales diagramas de secuencia que se utilizaron en la implementación de la aplicación. También se incluye una pequeña sección de arquitectura donde se retoman algunos puntos de arquitectura e implementación para dar una mejor visión de la solución construida. Posteriormente se hace una descripción de las principales clases del diseño que se utilizaron en la implementación de la aplicación.

2.1 Arquitectura de la solución

La arquitectura del módulo de visualización de informes del proyecto SCADA-GALBA es una solución orientada al patrón arquitectónico **N-Capas** específicamente 3 capas.

Capítulo 2: Descripción y construcción de la solución propuesta.

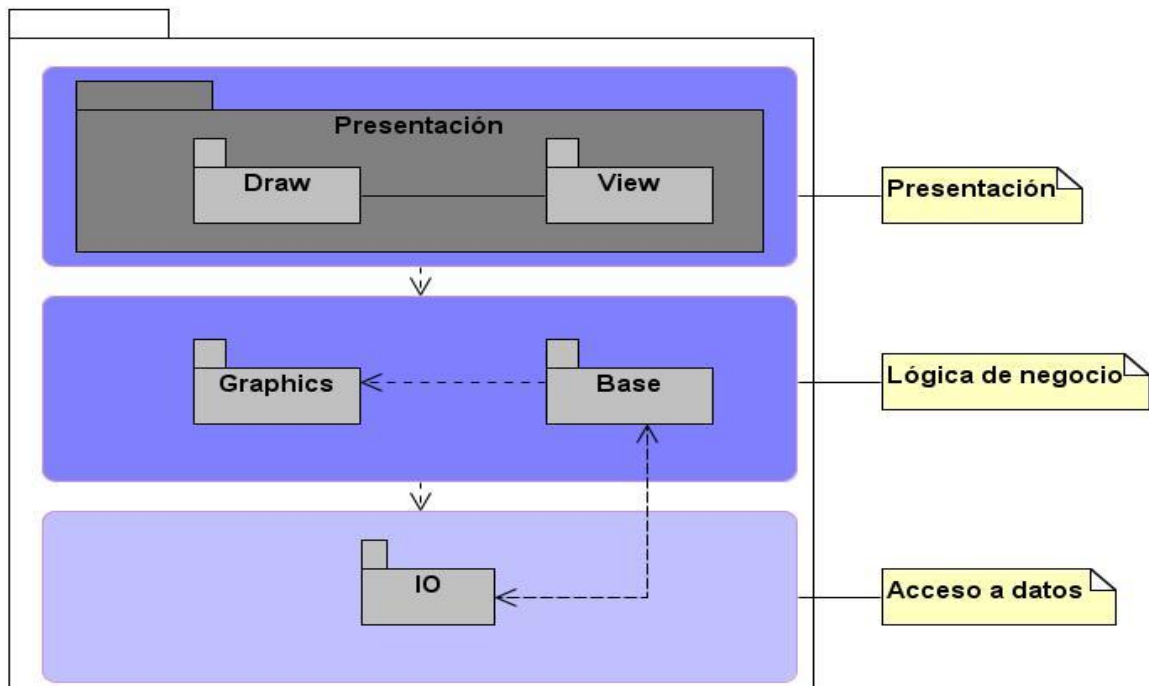


Fig. 9: Diagrama del patrón arquitectónico Tres Capas.

La arquitectura se encuentra representada por 3 capas lógicas que dan un alto nivel de encapsulamiento de las responsabilidades, permitiendo reducir al máximo el acoplamiento y aumentar la reutilización entre las mismas. Esta distribución de las capas permite que se realicen grandes cambios sin siquiera tener que realizar cambios en las demás capas una vez que estas tengan bien definidas la comunicación entre ellas.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, además de que se pueden en cada capa mezclar una serie de patrones arquitectónicos como es el caso de la capa de presentación donde se implementó el patrón Modelo Presentación el cual es una variante del patrón Modelo-Vista-Controlador (MVC). El patrón Modelo-Presentación se basa en separar las responsabilidades dentro de la aplicación en dos dominios

Capítulo 2: Descripción y construcción de la solución propuesta.

fundamentales. La lógica relacionada a la interacción con el usuario se inscribe en lo que se denomina Presentación, y la lógica perteneciente al negocio en el Modelo.

➤ **Capa de presentación**

Es la interfaz de comunicación de la aplicación con un usuario determinado. Está compuesta por todas las interfaces de usuario y los componentes necesarios para su correcto funcionamiento. Estos elementos pueden ser ficheros JavaScript, CSS, entre otros. Esta capa se encuentra representada por el proyecto web en el paquete WT de la aplicación, y tiene interacción directa con la de Negocio.

➤ **Capa de negocio**

Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

➤ **Capa de acceso a datos**

Es la capa que se encarga de acceder a los datos. Está formada por un gestor de bases de datos que realiza todo el almacenamiento de información, recibe solicitudes de almacenamiento, actualización o eliminación de información desde la capa de negocio.

2.2 Paquetes de diseño arquitectónicamente significativos

Esta sección describe la descomposición del sistema en subsistemas y módulos. Además, para cada subsistema su descomposición en clases y la descripción de sus responsabilidades así como las relaciones más importantes, operaciones y sus atributos.

Entre los principales paquetes se encuentran:

- Paquete Base.
- Paquete Graphics

Capítulo 2: Descripción y construcción de la solución propuesta.

- Paquete IO
- Paquete WT
 - Paquete Base
 - Paquete Draw.
 - Paquete View
- Paquete Types
- Paquete Utils

2.2.1 Paquete Base

En este paquete se encuentran las clases sobre las cuales se soportara parte del sistema, estas definen las principales interfaces que se implementarán para la visualización de los reportes, fabricación y renderizado de los objetos y el control de la aplicación en general. Como principales elementos de este paquete se encuentran las clases Document, Engine, Report, RenderDetailSection y RenderSection encargadas de crear los objetos, administrar la entrada/salida de los datos, administrar las plantillas de reportes y visualizar.

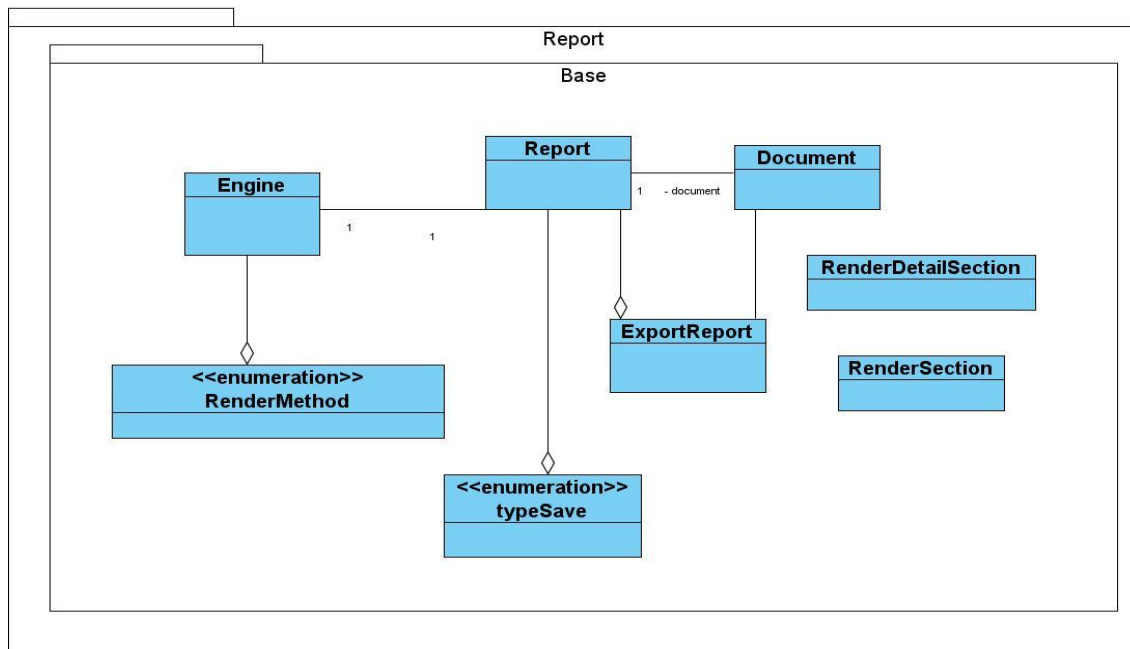


Fig. 10. Diagrama de clases. Paquete Base. [Ver descripción](#)

Capítulo 2: Descripción y construcción de la solución propuesta.

2.2.2 Paquete Graphics

En este paquete se encuentran todas las clases entidades del generador, o sea, todas aquellas clases que representan objetos tanto visuales como no visuales en el proceso de generación y visualización de un reporte. Entiéndase por objetos visuales todos aquellos objetos que forman parte de la visualización de un reporte (Ejemplo: Label, Picture, Line,...) y no visuales, aquellos que no representan forma visual en el proceso de visualización pero representan datos necesarios e imprescindibles para su generación (Ejemplo: Variable, Parameter ...).

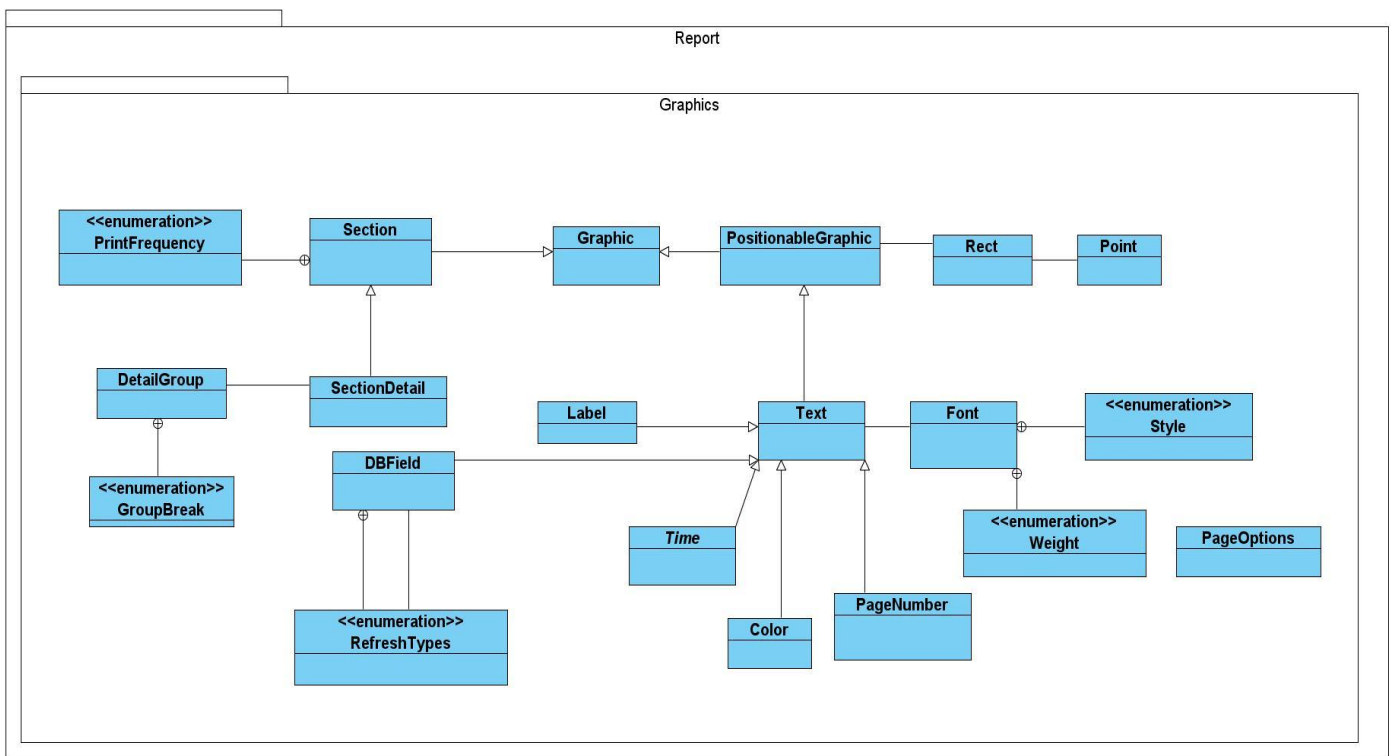


Fig. 11. Diagrama de Clases. Paquete Graphics. [Ver descripción](#)

2.2.3 Paquete WT

En el paquete WT se modelan todas las clases que tienen algún tipo de dependencia con la biblioteca WT utilizada para implementar el visualizador de informes, esto se hace para lograr independencia total de las vistas con la lógica y las entidades del sistema. La información que se maneja dentro de dicho paquete se encuentra organizada en dos paquetes; el paquete Draw, y el paquete View.

Capítulo 2: Descripción y construcción de la solución propuesta.

2.2.3.1 Paquete View

En este paquete se encuentran las clases con las que se construyen las interfaces gráficas de usuarios. La clase que implementa la interfaz principal ReportView, así como el resto de los componentes que colaboran para conformar las interfaces con las que el usuario va a interactuar.

2.2.3.2 Paquete Draw

En este paquete se encuentran un grupo de clases que permiten la visualización de cada entidad del generador, que intervenga en la visualización de un reporte, sea la de todas las mencionadas en el subsistema Graphics que eran objetos visuales.

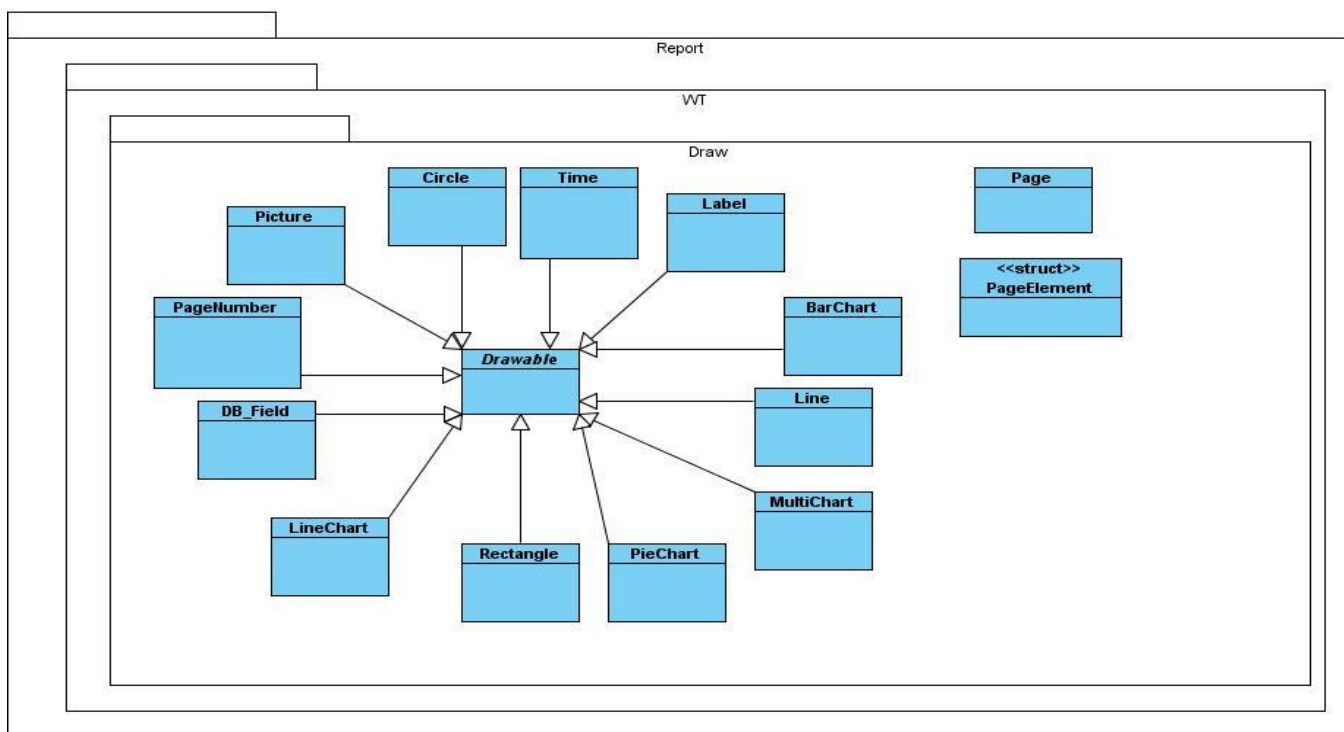


Fig. 12. Diagrama de clases. Paquete Draw. [Ver descripción](#)

2.2.4 Paquete IO (Input/Output)

Capítulo 2: Descripción y construcción de la solución propuesta.

En este paquete se encuentra todas aquellas clases que permiten el manejo de todo lo referente a adquisición y procesamiento de datos de Bases Datos en el reporte que se genere y visualice en un momento dado (Ejemplo: Connection, Query, IOManager).

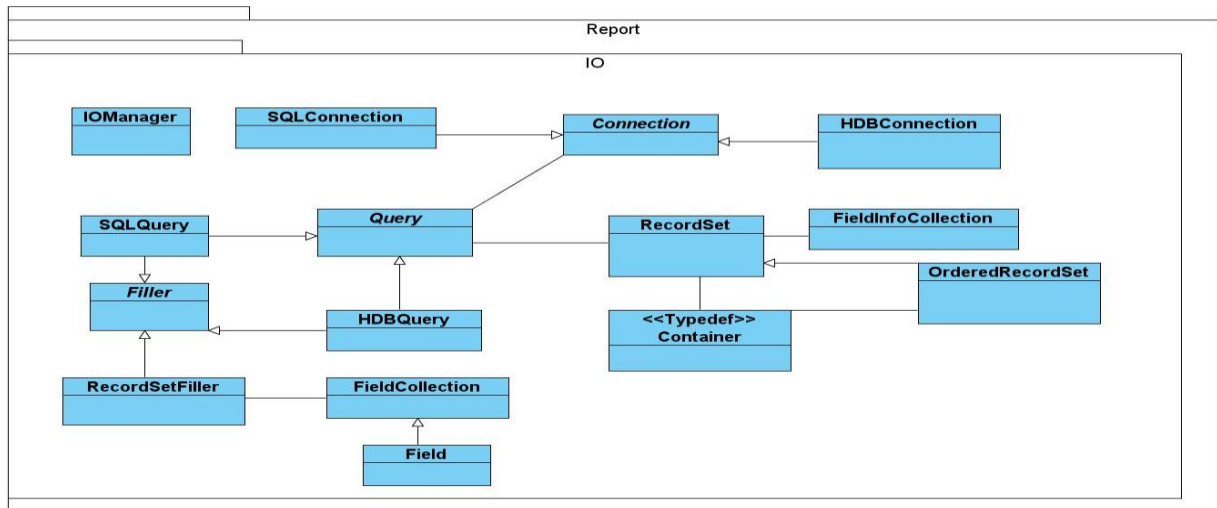


Fig. 13. Diagrama de clases. Paquete IO. [Ver descripción](#)

2.2.5 Paquete Exceptions

En este paquete se encuentra un grupo de clases que permiten el manejo de excepciones dentro del proceso de generación y visualización de reportes.

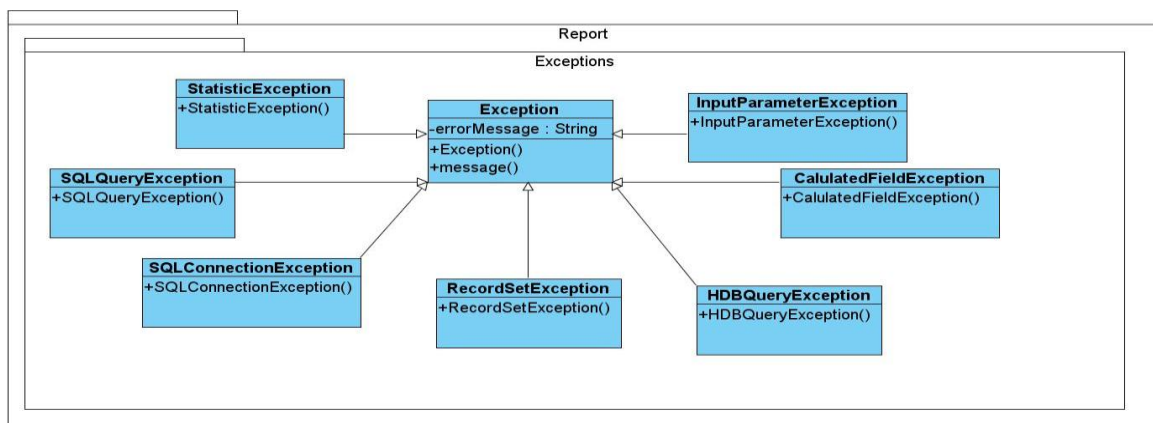


Fig. 14. Diagrama de clases. Paquete Exceptions. [Ver descripción](#)

Capítulo 2: Descripción y construcción de la solución propuesta.

2.2.6 Paquete Utils

En este paquete se encuentran las clases que se conectan a la base de datos como son (HDBConnectionConfig, SQLConnectionConfig y MultiSortedContainer).

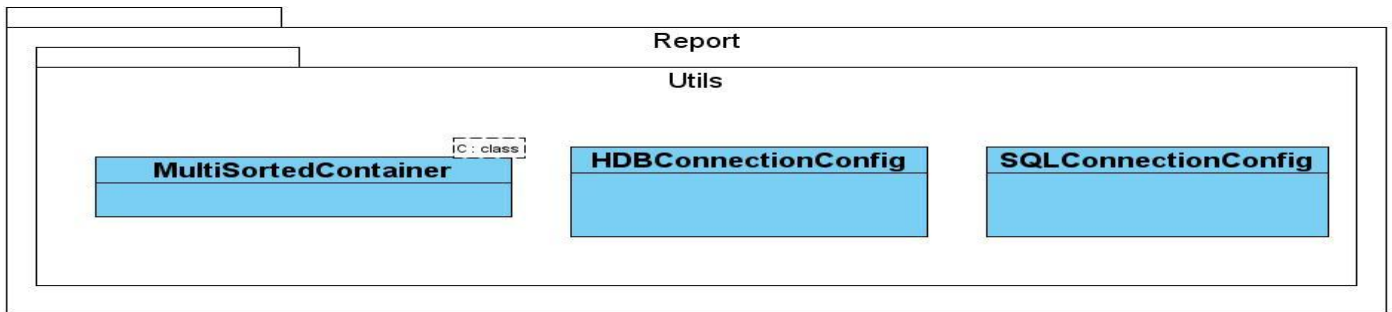


Fig. 15. Diagrama de clases. Paquete Utils.

2.2.7 Paquete Types

En este paquete se encuentran las clases que redefinen los tipos de datos, que resultarían más familiares al programador, ejemplo de las mismas son: Name, PageType, String, Variant, Types.

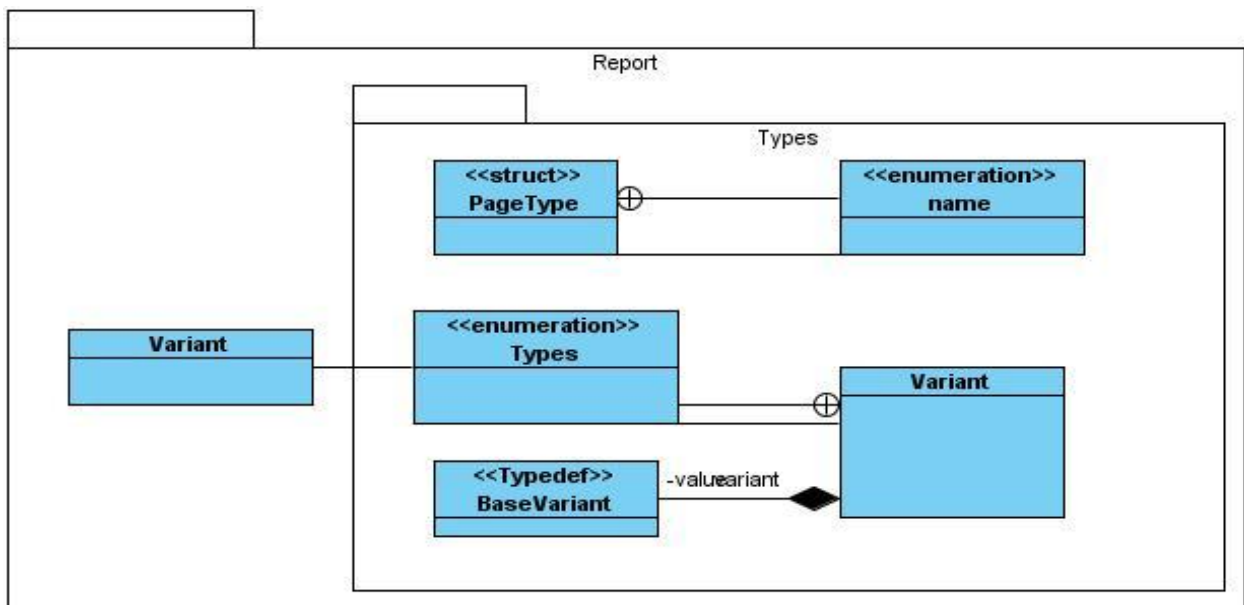


Fig. 16. Diagrama de clases. Paquete Types.

Capítulo 2: Descripción y construcción de la solución propuesta.

2.3 Patrones

“En la terminología de objetos, el patrón es una descripción de un problema y su solución que recibe un nombre y que puede emplearse en otros contextos; en teoría, indica la manera de utilizarlo en circunstancias diversas. Muchos patrones ofrecen orientación sobre cómo asignar las responsabilidades a los objetos ante determinada categoría de problemas. (...) Los patrones no se proponen descubrir ni expresar nuevos principios de la ingeniería de software. Todo lo contrario: intentan codificar el conocimiento, las expresiones y los principios ya existentes: cuanto más trillados y generalizados, tanto mejor.” (21)

Dada la similitud existente entre el subsistema que se necesita y una aplicación de gestión, se propone una arquitectura en capas, la cual da una gran flexibilidad a la aplicación dado el bajo acoplamiento funcional que se logra con la misma, permitiendo a la aplicación adaptarse de forma relativamente fácil a cambios (adiciones, modificaciones o eliminaciones).

En el diseño de la aplicación se aplicaron en todos los paquetes patrones GRASP y GOF, los GRASP son Patrones Generales de Software para Asignación de Responsabilidades considerados como "Buenas Prácticas" en el diseño de software. Los patrones GOF son patrones de diseño.

Los patrones de diseño tienen diferentes clasificaciones en dependencia de su funcionalidad.

- **Patrones Creacionales:** Muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones, normalmente serán resueltas dinámicamente, decidiendo qué clases instanciar o sobre qué objetos u objeto delegará responsabilidades.
- **Patrones estructurales:** Describen la forma en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros.
- **Patrones de comportamiento:** Se utilizan para organizar, manejar y combinar comportamientos.

Dentro de los patrones **creacionales** que se utilizaron en el diseño de la aplicación podemos destacar:

Capítulo 2: Descripción y construcción de la solución propuesta.

Patrón **singleton**: este asegura que solo se pueda crear una instancia de la clase y ofrece un punto global de acceso a esta instancia. El uso de este patrón permite que las conexiones a la BD solo puedan ser creadas sola una vez.

El funcionamiento de este patrón es muy sencillo y podría reducirse a los siguientes conceptos:

1. Ocultar el constructor de la clase Singleton, para que los clientes no puedan crear instancias.
2. Declarar en la clase Singleton una variable miembro privada que contenga la referencia a la instancia única que queremos gestionar.
3. Proveer en la clase Singleton una función o propiedad que brinde acceso a la única instancia gestionada por el Singleton. Los clientes acceden a la instancia a través de esta función o propiedad.

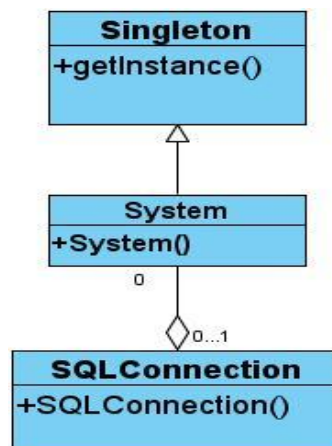


Fig. 17: Diagrama del patrón Singleton.

Dentro de los patrones **GRASP** que se utilizaron en el diseño de la aplicación podemos destacar:

Patrón **Creador**: el patrón creador nos ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

La nueva instancia deberá ser creada por la clase que:

- Tiene la información necesaria para realizar la creación del objeto, o
- Usa directamente las instancias creadas del objeto, o

Capítulo 2: Descripción y construcción de la solución propuesta.

- Almacena o maneja varias instancias de la clase
- Contiene o agrega la clase.

Este patrón es utilizado en la clase Document, ya que es la responsable de crear las secciones por las que esta compuesto un Reporte, dígame Encabezado y pie de reporte, Encabezado y pie de página y los detalles.

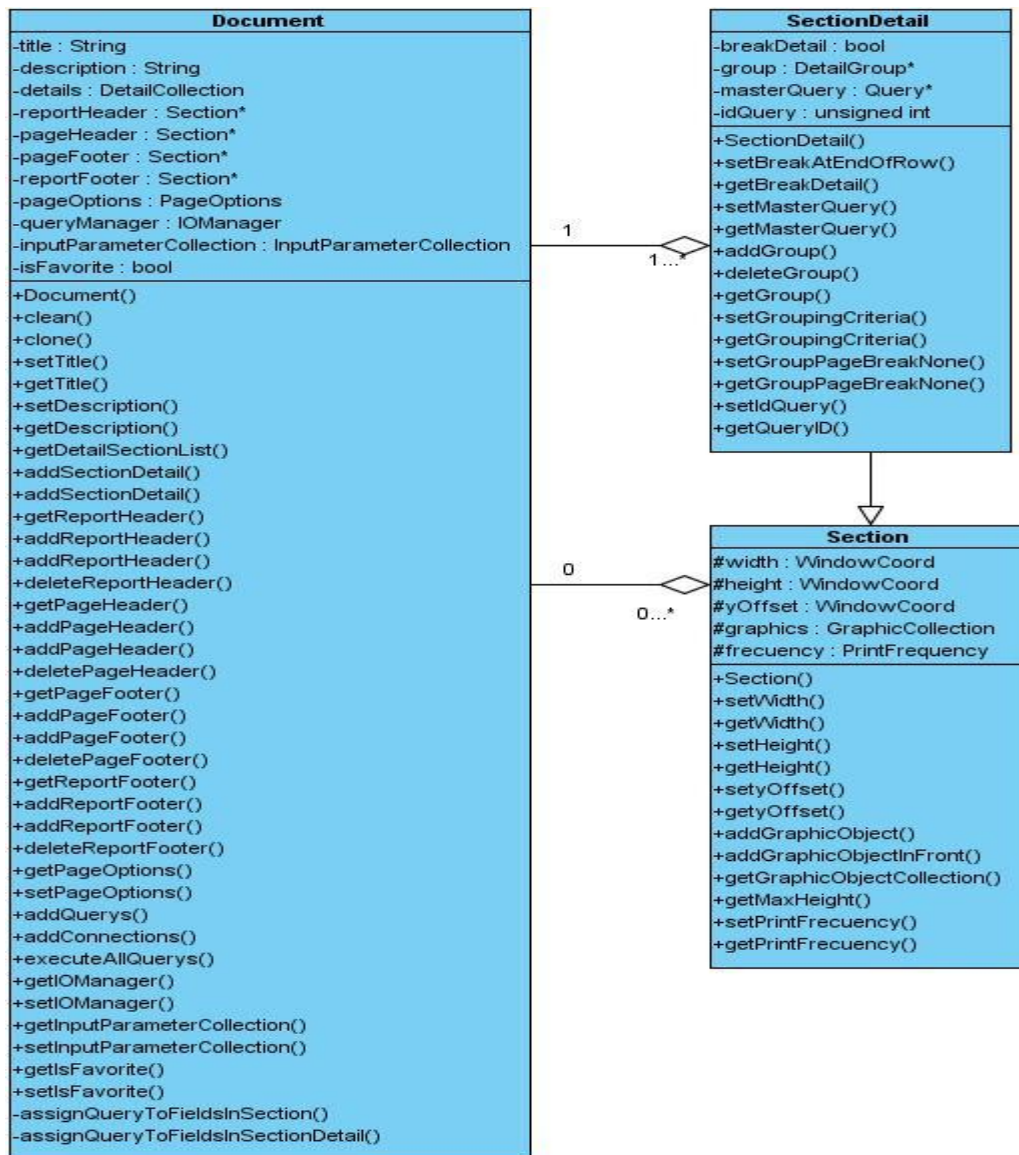


Fig. 18: Diagrama del patrón Creador.

Capítulo 2: Descripción y construcción de la solución propuesta.

Patrón **controlador**: el patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control.

Fue implementado en el generador de reportes en la clase Report que es la encargada de recibir los datos enviados por el usuario como son las opciones de procesamiento del reporte dígase opciones de previsualización, y de almacenamiento.

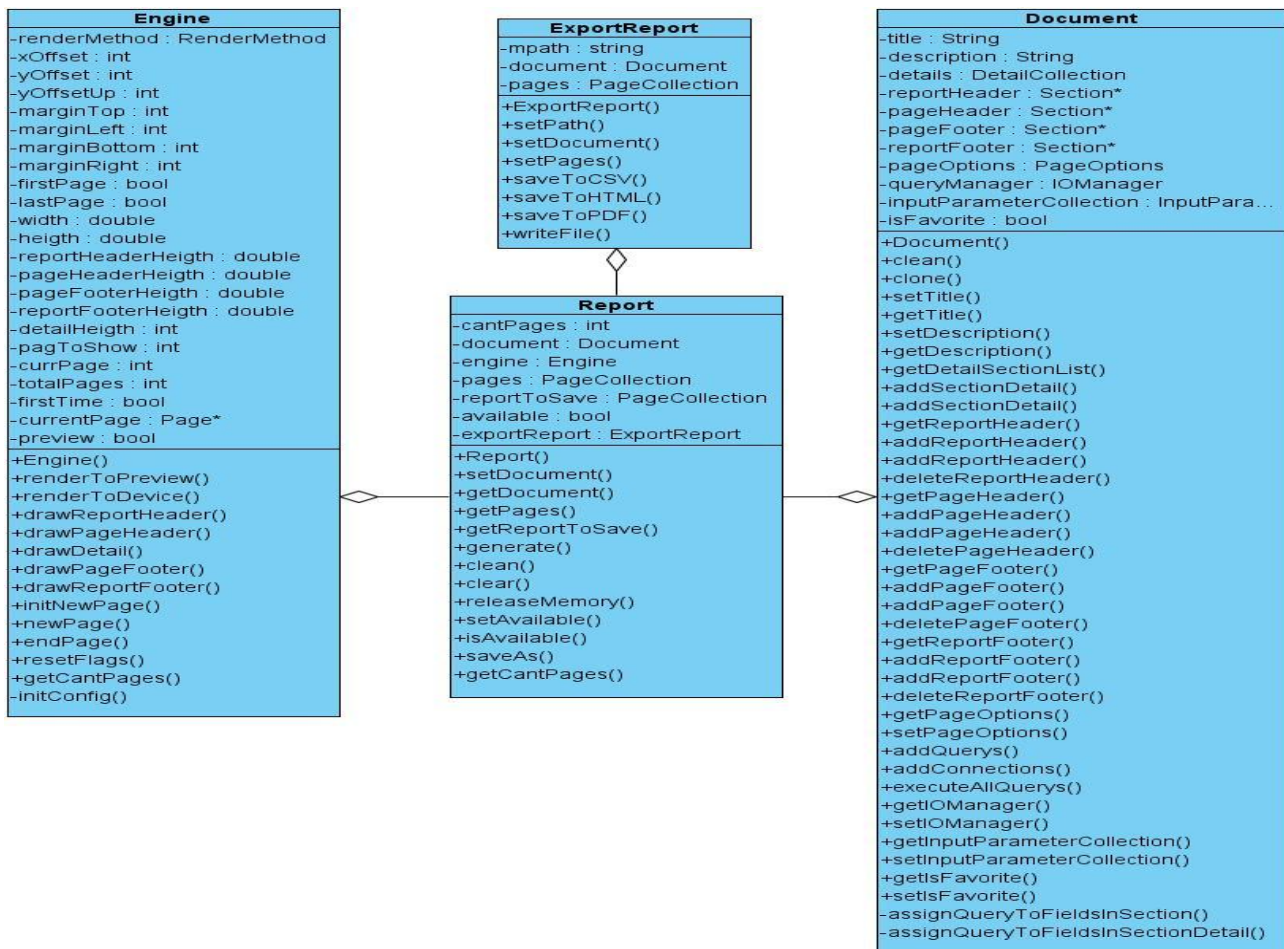


Fig. 19: Diagrama del patrón Controlador.

Capítulo 2: Descripción y construcción de la solución propuesta.

2.4 Principales funcionalidades.

A continuación se presenta la explicación de las principales funcionalidades del generador de informes, para ello se muestran los diagramas de clases y de secuencia de las mismas.

2.4.1 Visualizar Informes

Permite al usuario la visualización de un informe determinado, el usuario escoge la opción de generar el informe. El generador de informes haciendo uso de la clase "Report" hace una llamada al método "generate" con sus respectivos parámetros, dándole la orden a la instancia de la clase "Engine" que contiene el generar un informe mediante el método "renderToPreview" el cual es el encargado de generar solamente la página que solicitó el usuario para optimizar la memoria.

Para ello las principales clases involucradas se muestran en el siguiente Diagrama de Clases.

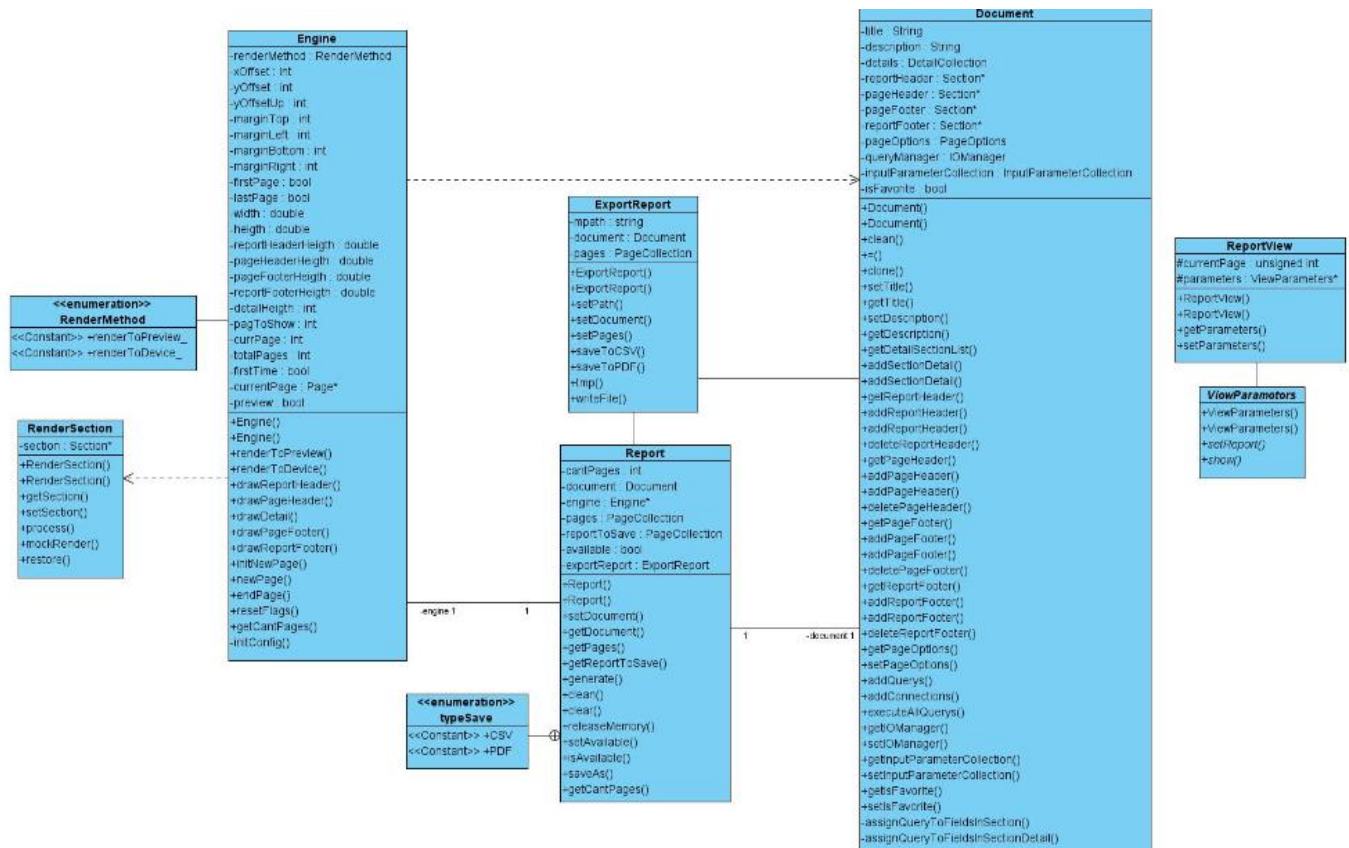


Fig. 20: Diagrama de clases Visualizar Informe.

Capítulo 2: Descripción y construcción de la solución propuesta.

2.4.2 Exportar informes

Los informes mostrados en la interfaz de visualización podrán ser guardados en disco duro, para cumplimentar esta funcionalidad, se dispone en el menú Archivo, la opción: Guardar y Guardar como. Si se selecciona la opción guardar, el informe será guardado con el nombre y en el lugar establecido con anterioridad, en caso de que el reporte no se haya guardado con anterioridad, se levantará una interfaz para especificar estos datos. Y si se selecciona la opción Guardar como, se podrá guardar el informe a formato PDF o CSV, especificándole el lugar donde se quiere guardar.

A esta opción se tendrá acceso de una forma directa a través del botón correspondiente colocado en la barra de herramientas ([Anexo 1](#)).

2.5 Conclusiones

En este capítulo se ha realizado la descripción de la mayor parte de los elementos implementados, se han seleccionado las funcionalidades más significativas ya implementadas y se han descrito detalladamente los principales diagramas que intervienen en su desarrollo.

Capítulo 3 **Validación de la solución.**

Introducción

Para la puesta en práctica de cualquier sistema complejo es de vital importancia probarlo para verificar su correcto funcionamiento. Muchos grupos de trabajo se limitan a dedicar una o dos personas a probar el sistema. En este capítulo se realiza el proceso de pruebas para validar el correcto funcionamiento de la solución propuesta. Aunque estas pruebas no garantizan la inexistencia de defectos en la aplicación sí pueden demostrar que existen errores en el software y así posteriormente poderlos corregir.

3.1 Pruebas

Una prueba es una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados y una evaluación es hecha de algún aspecto del sistema o componente.

Los diferentes niveles de pruebas son:

- **Nivel de unidad:** Está enfocada al código fuente de los componentes y verifica todos los flujos de control. Este tipo de prueba pasa primero por la revisión del programador.
- **Nivel de integración:** Prueba los componentes combinados para ejecutar casos de uso. Tiene como objetivo descubrir errores en las especificaciones de las interfaces de las clases.
- **Nivel de sistema:** Prueba el software funcionando como un todo. Aceptable cuando el software se encuentra en fase de construcción.
- **Nivel de aceptación:** Es la prueba final antes del despliegue del sistema. Generalmente las realizan los usuarios finales y es llamada prueba piloto.

3.1.1 ¿Qué es una Prueba de Unidad?

Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Las librerías de pruebas de unidad formalizan este trabajo al proporcionar clases para pruebas.

La prueba de unidad ayuda a que el módulo se haga independiente. Esto quiere decir que un módulo que tiene una prueba de unidad se puede probar independientemente del resto del sistema.

Una buena prueba de unidad:

- No requiere de mucho código para prepararla: Una prueba de unidad cuya preparación toma 3 minutos (conectándose a la base de datos, levantando un servidor de Web, etcétera) se debe considerar demasiado pesada para ejecutarse cada vez que se hagan cambios.
- Prueba sólo un método. Aunque no siempre es necesario hacer "una prueba por método", sí es importante asegurar que se cubra toda la clase. Muchas veces se requiere más de un método de prueba por cada método en la clase, para poder probar diferentes datos y casos extremos.
- Verifica los resultados con aserciones. De preferencia se debe de chequear sólo un resultado. Se pueden chequear varias partes del mismo resultado, pero diferentes condiciones deben probarse en un método separado.
- Deja los datos de la misma manera en que los encontró. Cuando la prueba comienza, los datos se preparan si es necesario, pero cuando la prueba termina, es importante que el sistema borre los datos después de la verificación. Es por esto que muchas librerías de pruebas unitarias tienen métodos llamados `setup()` y `teardown()` o similares. (1)

3.2 Diseño de las Pruebas de Unidad

Para la realización de las pruebas se utilizan como recursos físicos: computadoras con un microprocesador Core Duo con una velocidad del CPU de 2.16 Ghz, la memoria RAM que tiene es de 1 Gb y un disco duro con capacidad de 100 Gbytes. Los recursos lógicos con los que se realizaron las pruebas fueron: sistema operativo Debian GNU/Linux con núcleo 2.6.21, el IDE de desarrollo es el Eclipse 3.2 con el plugin CDT 2.1 para programar en C++, y el framework para automatizar las pruebas es el CxxTest, que se integra con el entorno de desarrollo y es favorable para el lenguaje que se utilizó. Todos los casos de pruebas que han sido realizados son del tipo Pruebas de Unidad y tienen como objetivo aislar el código fuente y validar el correcto funcionamiento de los métodos que se implementan. Se realizaron por las principales funcionalidades y las clases dentro del código correspondiente.

3.3 Pruebas de clases dentro del método "Visualizar Informe".

3.3.1 Clase TestEngine

Capítulo 3: Validación de la solución.

Casos de prueba del método: **void Engine::newPage(Draw::PageCollection& pages, Document& document, RenderSection& renderSection)**

Variables a considerar en el caso de prueba: **PageCollection& pages, Document& document, RenderSection& renderSection.**

Clase de Equivalencia para la variable: **pages**

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Valores que indican el número de las páginas.	Válida

Clase de Equivalencia para la variable: **document**

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Variable que representa el informe al cual se le va a adicionar la nueva página.	Válida

Clase de Equivalencia para la variable: **renderSection**

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Variable que permite dibujar la lista de componentes de una sección.	Válida

Casos de Prueba:

1-void TestaddPages_1 ();

Caso de prueba	Objetivo de la Prueba	Datos de entrada	Salida Esperada	Salida Obtenida	Observación
1	Verificar la adición de la	Un objeto de tipo PageCollection&	Adición de una nueva	Página adicionada	

Capítulo 3: Validación de la solución.

	página al ejecutar el método	distinto de NULL	página		
2					

Casos de prueba del método: **int Engine::getCantPages()**

Variables a considerar en el caso de prueba: **int totalPages**

Clase de Equivalencia para la variable: **totalPages**

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Valores enteros que indiquen la cantidad de páginas total	Válida

3.3.2 Clase TestReport

Casos de prueba del método: **void setAvailable (bool available)**

Variables a considerar en el caso de prueba: **bool available**

Clase de Equivalencia para la variable: **available**

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Verificar que cambie el valor del atributo available.	Válida

Casos de Prueba:

1-void testSetAvailable ()

Caso de Prueba	Objetivo de la Prueba	Datos de Entrada	Salida Esperada	Salida Obtenida	Observación
1	Verifica que se pueda cambiar el valor del atributo.	false	false	false	

Capítulo 3: Validación de la solución.

Casos de prueba del método: **void clean()**

Variables a considerar en el caso de prueba: **PageCollection pages**

Clase de Equivalencia para la variable: **pages**

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Verificar que la colección de página quede vacía.	Válida

Casos de Prueba:

1-void testClean()

Caso de Prueba	Objetivo de la Prueba	Datos de Entrada	Salida Esperada	Salida Obtenida	Observación
1	Comprobar que la colección de páginas quede vacía.	PageCollection pages	0	0	

3.3.3 Clase TestDocument

Casos de prueba del método: **void setName (Name name)**

Variables a considerar en el caso de prueba: **Name name**

Clase de Equivalencia para la variable: **name**

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Verificar que cambie el valor del atributo name.	Válida

Casos de Prueba

1-void testSetName()

Caso	Objetivo de la	Datos de entrada	Salida	Salida	Observación
------	----------------	------------------	--------	--------	-------------

Capítulo 3: Validación de la solución.

de prueba	Prueba		Esperada	Obtenida	
1	Verificar que se pueda cambiar el valor del atributo	Cadena pasada como parámetro.	El la misma cadena pasada en los datos de entrada	El la misma cadena pasada en los datos de entrada	

Casos de prueba del método: ***void deletePageHeader()***

Variables a considerar en el caso de prueba: ***Section * pPageHeader***

Clase de Equivalencia para la variable: ***pPageHeader***

Clase de Equivalencia para la variable: ***pPageHeader***

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Valores correctos para un objeto de tipo Section.	Válida

Casos de Prueba

1-void testDeletePageHeader_1 ()

2-void testDeletePageHeader_2 ()

Caso de prueba	Objetivo de la Prueba	Datos de entrada	Salida Esperada	Salida Obtenida	Observación
1	Verificar que el método funcione para una variable distinta de 0.	Un objeto de tipo pPageHeaderdistinto de NULL	El sistema debe borrar la dirección de memoria de la variable especificada	El sistema borró la dirección de memoria de la variable especificada	
2	Verificar que el método funcione para una variable nula.	Un objeto de tipo pPageHeader igual NULL.	Que el sistema no ejecute ninguna acción.	El sistema no ejecutó ninguna acción.	

Casos de prueba del método: **void Document::addSectionDetail(Graphics::SectionDetail***

sectionDetail) Variables a considerar en el caso de prueba: **SectionDetail* sectionDetail**

Clase de Equivalencia para la variable: **sectionDetail**

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Valores correctos para una objeto de tipo Section.	Válida

Caso de Prueba

1-void testAddSectionDetail _1 ()

2-void testAddSectionDetail _2 ()

Caso de prueba	Objetivo de la Prueba	Datos de entrada	Salida Esperada	Salida Obtenida	Observación
1	Verificar que se La variable pueda ser adicionada	SectionDetail* sectionDetail = new sectionDetail();	Que el sistema adicione correctamente la variable.	El sistema adicionó correctamente la variable.	
2	Verificar que la pila sea distinta de NULL.	push_back(sectionDetail);	Que el sistema adicione correctamente a la pila.	El sistema adicionó correctamente la variable a la pila.	

3.4 Pruebas a las clases dentro del método “Exportar Informe a CSV”.

3.3.4 Clase ExportReport

Casos de prueba del método: **void ExportReport::saveToCSV()**

Variables a considerar en el caso de prueba: **document**

Clase de Equivalencia para la variable: **document**

Capítulo 3: Validación de la solución.

Clase de Equivalencia	Clase de Equivalencia	Clasificación de las Clases de Equivalencia
1	Verifica que la variable no sea nula	Válida

Casos de Prueba:

1-void testIsEmpty()

Caso de Prueba	Objetivo de la Prueba	Datos de Entrada	Salida Esperada	Salida Obtenida	Observación
1	Verificar que el método funciona para una variable de tipo document.	Document * document = new document();	Asignación correcta de las páginas a exportar.	Asignación correcta de la colección de páginas	

3.5 Resumen de Pruebas.

En el proceso de pruebas efectuado se diseñaron 8 Casos de Prueba a nivel de unidad con el objetivo de verificar el correcto funcionamiento de la aplicación y su cumplimiento con las principales funcionalidades. Se obtuvo un total de 5 no conformidades, las cuales fueron solucionadas. Como resultado final se logró una aplicación que cumple con los requisitos funcionales. No se asegura el cumplimiento de los requisitos no funcionales dado que el producto no fue sometido a pruebas para la verificación de los mismos.

3.6 Conclusiones

En este capítulo se realizaron pruebas al sistema con el objetivo de verificar el correcto funcionamiento de la aplicación. Con la culminación del mismo se puede concluir que se cumplió con los objetivos propuestos: detectar no conformidades en el sistema tras la puesta en práctica de los casos de prueba diseñados y dar solución a las no conformidades encontradas. Es necesario destacar que las pruebas realizadas validan el correcto funcionamiento de la solución atendiendo a los requisitos funcionales de la misma.

Conclusiones

- Se hizo un estudio de la bibliografía en lo referente al estado del arte de las principales tendencias y tecnologías usadas en el desarrollo de generadores de reportes para la web.
- El paradigma de software libre facilitó en gran medida la reutilización de código y un desarrollo acelerado de la aplicación.
- La arquitectura N-Capas específicamente Tres Capas es muy conveniente para el desarrollo del módulo de visualización de informes, ya que permite reducir al máximo el acoplamiento y aumentar la reutilización entre las capas.
- La implementación basada en el análisis y diseño, respetando la arquitectura propuesta, logró una aplicación robusta permitiendo así la validación de las funcionalidades exigidas en los requisitos.
- Las pruebas unitarias ofrecieron valiosa información que ayudó a la corrección de los errores identificados y a implementar un producto con mayor calidad y en condiciones favorables para continuar su implementación.

Recomendaciones

- Continuar el desarrollo de nuevas funcionalidades del generador de informes del SCADA.
- Se recomienda finalizar la funcionalidad: imprimir informes, ya que la biblioteca por ser relativamente joven, no ofrecía facilidades para llevar a cabo dicha funcionalidad.
- Evaluar la posibilidad del uso de la implementación del módulo de visualización web de informes en otros sistemas en el plano nacional utilizando las funcionalidades descritas en el presente trabajo.

Referencias Bibliográficas

1. **Sehara, Yossel Luis.** *Implementación de un modelo para la configuración de un sistema SCADA.* Ciudad de La Habana : julio, 2008.
2. **Osmany Jorge Riverón, Hilario Calleja.** *Modelación de un reporteador para sistemas SCADA.* Ciudad de la Habana : s.n., 2006.
3. **Bernardo Zaragoza Hijuelos, Raudi Agdel Bacallao.** *Implementación de un módulo de generación de reportes para sistemas de supervisión, control y adquisición de datos.* . Ciudad de La Habana : julio, 2008.
4. **SCADA.** *Manual de usuario para explotación del módulo generador de reporte.* Ciudad de La Habana : s.n., 2007.
5. Stimulsoft Reports.Web. [En línea] 2010. [Citado el: 30 de 01 de 2010.] <http://www.stimulsoft.com/es/ReportsWeb.aspx>.
6. XML Report Generator (XMLRG). [En línea] 2005-2010. <http://www.renderx.com/tools/xmlrg.html>.
7. **Ltda.,** **Connectividad.net.** *Connectividad.net.* [En línea] 2008. <http://www.connectividad.net/default.asp>.
8. ABARTIA TEAM. [En línea] 2008. <http://www.abartiateam.com/jasperreports>.
9. **Ramírez, Oilede Pérez y López, Yonisley García.** *Desarrollo del Componente Trabajador del Subsistema Capital Humano del sistema Cedrux.* Ciudad de La Habana : s.n., 2009.
10. El CoDiGo K . [En línea] 26 de febrero de 2010. <http://www.elcodigok.com.ar/2010/02/codeblocks-el-ide-para-c/>.
11. Hosting Peru. [En línea] diciembre de 2009. <http://www.hosting-peruano.com/protocolo-de-transferencia-de-hipertexto-http/>.
12. **Henst, Christian Van Der.** *maestros del web.* [En línea] 1997. <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>.
13. **Wt, C++ Web Toolkit.** *Solutions for web-based systems.* [En línea]. <http://www.webtoolkit.eu/wt>.
14. Tigris.org. [En línea] 2009. <http://cxxtest.tigris.org/>.
15. HERRAMIENTAS CASE . [En línea] 2008. <http://herramientas-case.blogspot.com/>.
16. **Calderón, Amaro.** *Metodologías Ágiles* . Perú : s.n., 2007.
17. Free Download Manager. [En línea] 2007. [http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(Mí\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(Mí)_14720_p/).

Referencias bibliográficas

18. **Sáez, Pedro Antonio Fernández.** Un Análisis Crítico de la Aproximación Model-Driven Architecture. E-Prints Complutense. [En línea] 2009. http://eprints.ucm.es/9880/1/Un_An%C3%A1lisis_Cr%C3%ADtico_Sobre_la_Aproximaci%C3%B3n_Model-Driven_Architecture.pdf.
19. Rational Unified Process. *Rational Unified Process*. [En línea] 2007. <http://www.rational.com.ar/herramientas/rup.html>.
20. **L, JORGE LUIS CORDERO.** *METODOLOGIAS AGILES. PROCESO UNIFICADO AGIL (AUP)*. LA PAZ, EL ALTO – BOLIVIA : s.n.
21. **Craig, Larman.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. La Habana : Félix Varela, 2004.

Bibliografía

1. **Martínez, D.L. & Valesani, M.E.** “Comparación de Frameworks de Desarrollo de Páginas Web con AJAX”. 9 de julio de 2008. [En Línea]. Consultado el 19 de enero de 2010 de <http://www.sicuma.uma.es/sicuma/independientes/argentina08/LaRed-Eugenia/index.html>
2. “Wt, C++ Web Toolkit”. [En Línea]. Consultado en enero de 2010 desde <http://www.webtoolkit.eu/wt#/documentation>
3. “Tutorial de Java”. [En Línea]. Consultado el 16 de enero de 2010 de http://www.wikilearning.com/tutorial/tutorial_de_java-caracteristicas_de_java/3938-5
4. “Java, Web and Design. HTML 5 Canvas element”. 8 de junio de 2009. [En Línea]. Consultado el 25 de enero de 2010 desde <http://veerasundar.com/blog/2009/06/html-5-canvas-element/>
5. “{entre llaves}: HTML 5: Canvas”. 9 de junio de 2009. [En Línea]. Consultado el 25 de enero de 2010 desde <http://entrellaves.blogspot.com/2009/06/html-5-canvas.html>
6. “Patrones GRASP (Patrones de Software para la asignación General de Responsabilidades). Parte II”. 8 de mayo de 2007. [En Línea]. Consultado el 22 de febrero de 2010 desde <http://jorgesaavedra.wordpress.com/category/patrones-grasp/>
7. “Herramientas CASE”. 25 de septiembre de 2008. [En Línea]. Consultado el 20 de febrero de 2010 desde <http://herramientas-case.blogspot.com/>
8. Biblioteca Virtual de la Universidad de Cienfuegos. “Tendencias, tecnologías y herramientas utilizadas”. [En Línea]. Consultado el 15 de febrero de 2010 desde <http://www.eumed.net/libros/2010b/698/Tendencias%20tecnologias%20y%20herramientas%20utilizadas.htm>
9. “Visual Paradigm for UML”. [En Línea]. Consultado el 20 de enero de 2010 desde http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/

Anexos

Anexo 1 Funcionalidad “Botón Guardar”.



Anexo 2 Tabla Comparativa.

Framework	Características Esenciales	Ventajas	Desventajas
C++ Web Toolkit	<p>Es una librería C++ para el desarrollo de aplicaciones web interactivas.</p> <p>Soporta navegadores como (Firefox / Gecko, Internet Explorer, Safari, Chrome, Konqueror y Opera) y también navegadores planos de HTML.</p> <p>Tiene soporte integrado para Unicode (UTF8/UTF16).</p>	<p>Genera código XHTML.</p> <p>Utiliza el built-in httpd para un fácil desarrollo e implementación.</p> <p>Presenta una API simple para el servidor de eventos.</p> <p>Desarrolla aplicaciones web, como aplicaciones de escritorio.</p> <p>Proporciona un amplio conjunto de widgets, que funcionan independientemente de la disponibilidad de JavaScript (pero se benefician de la disponibilidad de JavaScript).</p> <p>Especificación única para la validación y el manejo de eventos tanto para el cliente como para el servidor.</p>	
ExtJS	<p>Es una librería JavaScript ligera y de alto rendimiento, compatible con la mayoría de los navegadores se utiliza</p>	<p>Permite la implementación de interfaces visuales muy potentes. Ofrece componentes para la implementación de tablas, árboles, formularios, y</p>	

	para crear páginas e interfaces web dinámicas.	contenedores. Admite diversos tipos de formatos como: XML, JSON, Arrays JavaScript. Puede ser adquirido bajo licencias libres y comerciales.	
Canvas HTML 5	Provee una superficie de dibujo con varias primitivas gráficas. Puede programarse, animarse y modificarse usando solo JavaScript.	Permite dibujar en la página y actualizar dinámicamente estos dibujos.	La falta de soporte de Explorer hace que todavía no sea muy recomendable su incorporación a las aplicaciones web.

Anexo 4 Descripción de clases. Paquete Base.

Nombre: Engine	
Descripción: Es la clase principal que se encarga de renderear sus datos de un documento para que sean visualizados en el visualizador del generador de reportes.	
Tipo de clase: Controladora	
Atributo	Tipo
firstPage	bool
lastPage	bool
width	double
heigth	double
reportHeaderHeigth	double
pageHeaderHeigth	double
pageFooterHeigth	double
reportFooterHeigth	double
detailHeigth	int
Para cada responsabilidad:	
Nombre:	void initConfig (Document& document)
Descripción:	Asigna valores a todos los atributos a partir de los datos del document.
Nombre:	void render (PageCollection& pages, Document& document)
Descripción:	Renderea document en una colección de pages para ser visualizado.
Nombre:	void drawReportHeader(Section*, RenderSection&)
Descripción:	Pinta un objeto de tipo Section en la sección ReportHeader de una página, utilizando un RenderSection.
Nombre:	void drawPageHeader(Section*, RenderSection&)
Descripción:	Pinta un objeto de tipo Section en la sección PageHeader de una página, utilizando un RenderSection.
Nombre:	void drawDetail(SectionDetail*, Document&, RenderDetailSection&)
Descripción:	Pinta un objeto de tipo SectionDetail en la sección Detail de una página, utilizando un RenderDetailSection.
Nombre:	void drawPageFooter(Section*, RenderSection&)
Descripción:	Pinta un objeto de tipo Section en la sección PageFooter de una página, utilizando un RenderSection.
Nombre:	void drawReportFooter(Section*, RenderSection&)
Descripción:	Pinta un objeto de tipo Section en la sección ReportFooter de una página, utilizando un RenderSection.
Nombre:	void newPage(PageCollection&, Document&, RenderSection&)
Descripción:	crea una nueva página en la colección de páginas para ser pintada
Nombre:	void endPage(Document&, RenderSection&)
Descripción:	crea la última página en la colección de páginas para ser pintada

Nombre: Document	
Descripción: Es la clase principal que contiene todos los objetos gráficos, gestiona la asignación de ellos y el manejo de los datos de entrada y salida a través del IOManager.	
Tipo de clase: Controladora	
Atributo	Tipo
title	String
description	String
details	DetailCollection
reportHeader	Section*
pageHeader	Section*
pageFooter	Section*
reportFooter	Section*
pageOptions	Graphics ::PageOptions
queryManager	IO::IOManager
inputParameterCollection	IO::InputParameterCollection
variableCollection	Graphics::VariableReportCollection
Para cada responsabilidad:	

Nombre:	void clean()
Descripción:	Limpia todos los datos de entrada y salida
Nombre:	void setTitle(String reportTitle)
Descripción:	Asigna un nuevo título al documento
Nombre:	String getTitle()
Descripción:	Devuelve el título del documento
Nombre:	void setDescription(String reportDescription)
Descripción:	Asigna una nueva descripción al documento
Nombre:	String getDescription()
Descripción:	Devuelve la descripción del documento
Nombre:	DetailCollection& getDetailSectionList()
Descripción:	Devuelve la lista de sección de detalles del documento
Nombre:	void addSectionDetail()
Descripción:	Adiciona una nueva sección de detalles no inicializada
Nombre:	void addSectionDetail(Graphics::SectionDetail* sectionDetail)
Descripción:	Adiciona una nueva sección de detalles pre-inicializada
Nombre:	Section* getReportHeader()
Descripción:	Devuelve el encabezado de reporte del documento
Nombre:	void addReportHeader(Section * pReportHeader)
Descripción:	Asigna un nuevo encabezado del reporte al documento
Nombre:	void addReportHeader()
Descripción:	Adiciona un nuevo encabezado del reporte no inicializado
Nombre:	Section* getPageHeader()
Descripción:	Devuelve el pie de página del documento

Nombre:	<code>void addPageHeader(Section * pPageHeader)</code>
Descripción:	Asigna un nuevo encabezado de página al documento
Nombre:	<code>void addPageHeader()</code>
Descripción:	Adiciona un nuevo encabezado de página no inicializado
Nombre:	<code>Section* getPageFooter()</code>
Descripción:	Devuelve el pie encabezado de página del documento
Nombre:	<code>void addPageFooter(Section * pPageFooter)</code>
Descripción:	Asigna nuevo pie de página al documento
Nombre:	<code>void addPageFooter()</code>
Descripción:	Adiciona un nuevo pie de página no inicializado
Nombre:	<code>Section* getReportFooter()</code>
Descripción:	Devuelve el pie de reporte del documento
Nombre:	<code>void addReportFooter(Section * pReportFooter)</code>
Descripción:	Asigna un nuevo pie de reporte al documento
Nombre:	<code>void addReportFooter()</code>
Descripción:	Adiciona un nuevo pie de reporte no inicializado
Nombre:	<code>Graphics::PageOptions& getPageOptions()</code>
Descripción:	Devuelve la configuración o opciones de una página
Nombre:	<code>void setPageOptions(Graphics::PageOptions& options)</code>
Descripción:	Asigna una nueva configuración o opciones de página
Nombre:	<code>void addQuerys(IO::Query* query)</code>
Descripción:	Adiciona una nueva consulta para manejo de entrada de datos
Nombre:	<code>void addConnections (IO::Connection* connection)</code>
Descripción:	Adiciona una nueva conexión para el manejo de entrada de datos
Nombre:	<code>void executeAllQuerys()</code>
Descripción:	Ejecuta todas las consultas y abre cada conexión
Nombre:	<code>IO::IOManager& getIOManager()</code>
Descripción:	Devuelve el controlador de datos de entrada-salida
Nombre:	<code>void setIOManager(IO::IOManager manager)</code>
Descripción:	Asigna un nuevo controlador de datos de entrada-salida
Nombre:	<code>IO::InputParameterCollection& getInputParameterCollection()</code>
Descripción:	Devuelve la colección de parámetros de entrada
Nombre:	<code>void setInputParameterCollection(inputParameterCollection)</code>
Descripción:	Asigna una nueva colección de parámetros de entrada
Nombre:	<code>Graphics::VariableReportCollection& getVariableCollection()</code>
Descripción:	Devuelve la colección de variables
Nombre:	<code>void setVariableCollection(variableCollection)</code>
Descripción:	Asigna una nueva colección de variables

Nombre: Report	
Descripción: Es la clase que se encarga de controlar y que contiene al documento.	
Tipo de clase: Controladora	
Atributo	Tipo
document	Document
engine	Engine
pages	PageCollection
available	bool
cantPages	int
reportToSave	PageCollection
exportReport	ExportReport
Para cada responsabilidad:	
Nombre:	void setDocument(Document& doc)
Descripción:	Asigna un nuevo documento
Nombre:	Document&getDocument()
Descripción:	Devuelve el atributo document
Nombre:	PageCollection&getPages()
Descripción:	Devuelve la colección de páginas
Nombre:	void generate()
Descripción:	Genera los datos del atributo document en la colección de páginas
Nombre:	void clean()
Descripción:	Limpia los datos de entrada-salida del documento
Nombre:	void clear()
Descripción:	Borra todas las páginas en la colección de página
Nombre:	void setAvailable(bool available)
Descripción:	Asigna nuevo valor al atributo available
Nombre:	bool isAvailable()
Descripción:	Devuelve el atributo available
Nombre:	Void saveAs()
Descripción:	Escoge el formato a salvar
Nombre:	Int getcantPages()
Descripción:	Devuelve la cantidad de páginas a salvar
Nombre:	Void releaseMemory()
Descripción:	Libera memoria, poniendo en cero la cantidad de páginas
Nombre:	PageCollection& getReportToSave()
Descripción:	Devuelve el reporte a exportar

Nombre: ExportReport
Descripción: Clase encargada de exportar los reportes a los diferentes formatos, tales como: CSV, HTML y PDF.

Tipo de clase: Controladora	
Atributo	Tipo
mpath	String
document	Document
pages	PageCollection
Para cada responsabilidad:	
Nombre:	void setPath(std::string mpath);
Descripción:	Asigna un nuevo directorio a salvar.
Nombre:	void setDocument(Document& document);
Descripción:	Asigna un nuevo documento para salvar.
Nombre:	void setPages(PageCollection& pages);
Descripción:	Asigna una nueva colección de páginas vacías para salvar a PDF.
Nombre:	void saveToCSV();
Descripción:	Método encargado de salvar los diferentes reportes al formato CSV.
Nombre:	void saveToHTML();
Descripción:	Método encargado de salvar los diferentes reportes al formato HTML.
Nombre:	void saveToPDF();
Descripción:	Método encargado de salvar los diferentes reportes al formato PDF.

Anexo 5 Descripción de clases. Paquete Graphics.

Nombre: SectionDetail	
Descripción: Clase que representa dentro de la sección es de un reporte la de detalles del mismo.	
Tipo de clase: Controladora	
Atributo	Tipo
breakDetail	bool
group	Base::DetailGroup*
masterQuery	IO::Query*
idQuery	unsigned int
Para cada responsabilidad:	
Nombre:	void setBreakAtEndOfRow(bool breakD)
Descripción:	Asigna un nuevo valor al atributo breakDetail
Nombre:	bool getBreakDetail()
Descripción:	Devuelve el atributo breakDetail
Nombre:	void setMasterQuery(IO::Query* mQuery)
Descripción:	Asigna una nueva consulta maestra
Nombre:	IO::Query* getMasterQuery()
Descripción:	Devuelve la consulta maestra
Nombre:	void addGroup()
Descripción:	Adiciona un nuevo grupo
Nombre:	Base::DetailGroup* getGroup()
Descripción:	Devuelve el grupo.
Nombre:	void setGroupingCriteria(IO::SortInfo criteria)
Descripción:	Asigna un nuevo criterio de agrupamiento
Nombre:	IO::SortInfo& getGroupingCriteria()
Descripción:	Devuelve un criterio de agrupamiento
Nombre:	void setGroupPageBreakNone(bool gB)
Descripción:	Asigna un nuevo valor al atributo groupBreak del atributo group
Nombre:	bool getGroupPageBreakNone()
Descripción:	Devuelve el atributo groupBreak del atributo group
Nombre:	void setIdQuery(unsigned int idQuery)
Descripción:	Asigna un nuevo Identificador de consulta maestra
Nombre:	unsigned int getQueryID()
Descripción:	Devuelve identificador de consulta maestra

Nombre: PageOptions	
Descripción: Clase que se encarga de configurar las propiedades de una página en un documento.	
Tipo de clase: Controladora	
Atributo	Tipo
resolution	ScreenResolution
marginTop	double

marginBottom	double
marginLeft	double
marginRight	double
pageSize	String
formats	map<String, PageType>
customWidth	double
customHeigh	double
width	double
height	double
orientationPortrait	bool
screenDPI	double
Para cada responsabilidad:	
Nombre:	void setFormat(String pageSize)
Descripción:	Asigna un nuevo formato de página
Nombre:	PageType getFormat()
Descripción:	Devuelve el formato de página
Nombre:	void setPageSize(String pagSize)
Descripción:	Asigna un nuevo tamaño de página
Nombre:	String getPageSize()
Descripción:	Devuelve tamaño de página
Nombre:	void setCustomPageSize(double width, double height)
Descripción:	Asigna nuevo tamaño por defecto de página
Nombre:	double getMarginTop()
Descripción:	Devuelve el ancho del margen superior
Nombre:	void setMarginTop(double v)
Descripción:	Asigna nuevo ancho de margen inferior
Nombre:	double getMarginBottom()
Descripción:	Devuelve el ancho del margen inferior
Nombre:	double getMarginBottominches()
Descripción:	Devuelve el ancho del margen inferior en pulgadas
Nombre:	void setMarginBottom(double v)
Descripción:	Asigna nuevo ancho de margen izquierdo
Nombre:	double getMarginLeft()
Descripción:	Devuelve el ancho del margen izquierdo
Nombre:	void setMarginLeft(double v)
Descripción:	Asigna nuevo ancho de margen
Nombre:	double getMarginRight()
Descripción:	Devuelve el ancho del margen derecho
Nombre:	void setMarginRight(double v)
Descripción:	Asigna nuevo ancho de margen derecho
Nombre:	double getCustomWidth()
Descripción:	Devuelve el ancho por defecto
Nombre:	double getWidthInInches()
Descripción:	Devuelve el ancho en pulgadas
Nombre:	void setCustomWidth(double v)
Descripción:	Asigna un nuevo ancho por defecto

Nombre:	double getCustomHeight()
Descripción:	Devuelve el largo por defecto
Nombre:	double getHeigthInInches()
Descripción:	Devuelve el ancho en pulgadas
Nombre:	void setCustomHeight(double v)
Descripción:	Asigna un nuevo largo por defecto
Nombre:	double getWidth()
Descripción:	Devuelve el ancho
Nombre:	void setWidth(double w)
Descripción:	Asigna nuevo ancho
Nombre:	double getHeigth()
Descripción:	Devuelve el largo
Nombre:	void setHeigth(double h)
Descripción:	Asigna nuevo largo
Nombre:	bool isPortrait()
Descripción:	Devuelve si tiene orientación Portrait
Nombre:	void setPortrait(bool yes)
Descripción:	Asigna orientación Portrait
Nombre:	double getSystemDPI()
Descripción:	Devuelve DPI del sistema

Nombre: Graphic	
Descripción: Clase que representa todos los objetos gráficos de reporte.	
Tipo de clase: Entidad	
Atributo	Tipo
name	Name
identifier	Identifier
Para cada responsabilidad:	
Nombre:	void setIdentier(Identifier iden)
Descripción:	Asigna un identificador
Nombre:	Identifier getIdentier()
Descripción:	Devuelve el identificador
Nombre:	void setName(Name n)
Descripción:	Asigna un nuevo nombre
Nombre:	Name getName()
Descripción:	Devuelve un nombre

Nombre: Text	
Descripción: Clase que representa todos los objetos con comportamiento de texto.	
Tipo de clase: Entidad	

Atributo		Tipo
font		Font
text		String
color		Color
Para cada responsabilidad:		
Nombre:	void setText(String text)	
Descripción:	Se le asigna un nuevo valor al atributo text.	
Nombre:	String getText() const	
Descripción:	Se retorna el valor del atributo text.	
Nombre:	Font getFont()	
Descripción:	Se retorna el valor del atributo font	
Nombre:	void setFont(SCADA:: Texts::Font)	
Descripción:	Se le asigna un nuevo valor al atributo font	
Nombre:	Void setColor(Color color)	
Descripción:	Asigna un nuevo color	

Nombre: Section	
Descripción: Clase que representa a una sección dentro de un reporte.	
Tipo de clase: Entidad	
Atributo	Tipo
width	WindowCoord
height	WindowCoord
yOffset	WindowCoord
frecuency	PrintFrequency
Para cada responsabilidad:	
Nombre:	void setWidth(WindowCoord)
Descripción:	Asigna nuevo ancho de la sección
Nombre:	WindowCoord getWidth()
Descripción:	Devuelve el ancho de la sección
Nombre:	void setyOffset(WindowCoord)
Descripción:	Asigna la posición del element con respect a la ventana
Nombre:	Utils::PrintFrequency getPrintFrecuency()
Descripción:	Devuelve la frecuencia de impresión

Anexo 6 Descripción de clases. Paquete Draw.

Nombre: Drawable	
Descripción: Clase que define la capacidad de un objeto de ser dibujable en un contexto.	
Tipo de clase: Interfaz	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	virtual void draw(Context cr, int yinit) = 0

Nombre: Page	
Descripción: Clase que define y visualiza el reporte, contiene el contexto (Context).	
Tipo de clase: Controladora	
Atributo	Tipo
context	Context
width	double
height	double
Para cada responsabilidad:	
Nombre:	void setContext(double w, double h)
Descripción:	Se le asigna un valor de ancho y un largo a la página, (width, height).
Nombre:	Context getContext()
Descripción:	Retorna el contexto.

Anexo 7 Descripción de clases. Paquete IO.

Nombre: IOManager	
Descripción: Clase que administra y maneja todos los datos de entrada y salida.	
Tipo de clase: Controladora	
Atributo	Tipo
queries	QueryCollection
connections	ConnectionCollection
Para cada responsabilidad:	
Nombre:	void exec()
Descripción:	Ejecuta las consultas y abre las conexiones
Nombre:	void addQuery(Query* query)
Descripción:	Adiciona una nueva consulta
Nombre:	Query* find(Identifier identifier)
Descripción:	Devuelve y busca la consulta con el identificador deseado
Nombre:	QueryCollection& getQueryCollection ()
Descripción:	Devuelve la colección de consultas
Nombre:	ConnectionCollection& getConnectionCollection ()
Descripción:	Devuelve la colección de conexiones
Nombre:	Connection* findConnection (Identifier identifier)
Descripción:	Devuelve y busca la conexión con el identificador deseado
Nombre:	void addConnection (Connection* connection)
Descripción:	Adiciona una nueva conexión
Nombre:	void clean()
Descripción:	Limpia todos los datos almacenados de entrada y salida
Nombre:	insertParametersValuesQuerys()
Descripción:	

Nombre: Connection	
Descripción: Interfaz encargada de proveer métodos para la conexión con la base de datos.	
Tipo de clase: Interfaz	
Atributo	Tipo
id	Identifier
Para cada responsabilidad:	
Nombre:	virtual bool isActive() = 0;
Nombre:	virtual void open () = 0;
Nombre:	virtual void close () = 0;
Nombre:	Identifier getIdentifier()
Nombre:	void setIdentifier(Identifier)
Nombre:	void isActive()
Nombre:	void open()
Nombre:	void close()

Nombre: Query	
Descripción: Clase que provee métodos para realizar consultas a base de datos.	
Tipo de clase: Controladora	
Atributo	Tipo
name	String
id	Identifier
connection	Connection
connectionId	Identifier
recordset	RecordSet
parameters	InputParameterCollection
sortCriteria	SortCollection
Para cada responsabilidad:	
Nombre:	String getName()
Descripción:	Devuelve el nombre de la consulta
Nombre:	void setName(String name)
Descripción:	Asigna nuevo nombre a la consulta
Nombre:	RecordSet* getResult()
Descripción:	Devuelve los datos de la consulta
Nombre:	InputParameterCollection& getParameterCollection ()
Descripción:	Devuelve la colección de parámetros de entrada
Nombre:	void setConnectionId(Identifier connectionId)
Descripción:	Asigna nuevo identificador de conexión
Nombre:	Identifier getConnectionId ()
Descripción:	Devuelve el identificador de la conexión
Nombre:	void setConnection (Connection* connection)
Descripción:	Asigna una nueva conexión
Nombre:	Connection* getConnection ()
Descripción:	Devuelve la conexión
Nombre:	SortCollection& getSortCriteria ()
Descripción:	Devuelve la colección de criterios de ordenamiento
Nombre:	void clear()
Descripción:	Limpia los datos de la consulta
Nombre:	virtual void execute() = 0

Nombre: RecordSet	
Descripción: Clase que almacena, organiza y manipula los datos que devuelve una consulta.	
Tipo de clase: Controladora	
Atributo	Tipo
info	FieldInfoCollection
data	RecordSetData

colerator	iterator
beginGroupIterator	iterator
pageSize	int
interposition	int
comparers	ComparerCollection
container	Container
sorts	SortCollection
Para cada responsabilidad:	
Nombre:	int recordCount()
Descripción:	Devuelve el número de filas
Nombre:	int fieldCount()
Descripción:	Devuelve el número de campos o columnas
Nombre:	FieldInfoCollection getFieldsInfo()
Descripción:	Devuelve el nombre de los campos o columnas
Nombre:	FieldCollection fields()
Descripción:	Devuelve todos los campos
Nombre:	FieldCollection fetchRow()
Descripción:	Devuelve todos los campos
Nombre:	FieldCollection fieldsByName(String columnName)
Descripción:	Devuelve el campo por nombre deseado
Nombre:	FieldCollection fieldsInGroupByColumnName(column, groupBegin)
Descripción:	Devuelve el campo por nombre deseado en un grupo
Nombre:	void moveFirst()
Descripción:	Mueve el puntero a la primera fila
Nombre:	void moveLast()
Descripción:	Mueve el puntero a la última fila
Nombre:	void moveNext()
Descripción:	Mueve el puntero a la siguiente fila
Nombre:	void movePrev()
Descripción:	Mueve el puntero a la anterior fila
Nombre:	int getPos()
Descripción:	Posición actual del puntero
Nombre:	bool end()
Descripción:	Determina si el puntero llegó al final
Nombre:	int getPageCount()
Descripción:	Cantidad de páginas a generar
Nombre:	bool pageEnd ()
Descripción:	Determina si el puntero está al final de una página
Nombre:	void sort ()
Descripción:	Ordena según un criterio las filas
Nombre:	int getIndexByField (const String& fieldName)
Descripción:	Posición de un campo determinado
Nombre:	void setSortCollection (SortCollection sorts)
Descripción:	Asigna una nueva colección de criterios de ordenamiento
Nombre:	SortCollection getSortCollection ()
Descripción:	Devuelve la colección de criterios de ordenamiento

Nombre:	void clear()
Descripción:	Borra todos los datos almacenados

Nombre: InputParameter	
Descripción: Clase que provee los datos para realizar consultas parametrizadas.	
Tipo de clase: Controladora	
Atributo	Tipo
value	Utils::Variant
information	string
Para cada responsabilidad:	
Nombre:	Utils::Variant getValue()
Descripción:	Devuelve el valor que presenta el parámetro
Nombre:	void setValue (Utils::Variant v)
Descripción:	Asigna un nuevo valor al parámetro
Nombre:	void setType(Utils::Variant::Types type)
Descripción:	Asigna un nuevo tipo de parámetro
Nombre:	Utils::Variant::Types getType()
Descripción:	Devuelve el tipo de parámetro
Nombre:	void setInformation(std::string information)
Descripción:	Asigna la información del parámetro
Nombre:	std::string getInformation()
Descripción:	Devuelve la información del parámetro

Anexo 8 Descripción de clases. Paquete Exceptions.

Nombre: Exception	
Descripción: Clase controladora y base en la jerarquía de clases para la manipulación de excepciones en Reportes. De la misma heredan las clases CalculatedFieldException, HDBQueryException, InputParameterException, SQLException, StaticticException...	
Tipo de clase: Controladora	
Atributo	Tipo
errorMessage	String
Para cada responsabilidad:	
Nombre:	String message()
Descripción:	Devuelve el mensaje de error

Glosario

-A-

ADO (ActiveX Data Objects): es uno de los mecanismos que usan los programas de computadoras para comunicarse con las bases de datos, darles órdenes y obtener resultados de ellas. Con ADO, un programa puede leer, insertar, editar, o borrar, la información contenida en diferentes áreas de almacenamiento dentro de la base de datos llamadas tablas.

Aserciones: La técnica de las aserciones consiste en sembrar el código de chequeos de integridad, de cosas que se suponen, de forma que si algo no es normal, se detecte lo antes posible, por sí mismo, en vez de tener que estar trazando marcha atrás las causas que han llevado a un fallo.

-C-

CGI (Common Gateway Interface): Interface Común de Pasarela. Interface de intercambio de datos estándar en WWW a través del cual se organiza el envío de recepción de datos entre visualizadores y programas residentes en servidores WWW.

-D-

Dispositivo de campo: Un dispositivo de campo es un sistema de transmisión de información (datos) que simplifica la instalación y operación de máquinas y equipamientos industriales utilizados en procesos de producción. El objetivo de un dispositivo de campo es sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control. Típicamente son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo como PLCs/PACs, transductores, actuadores y sensores.

-E-

ERP (Enterprise Resource Planning): sistema de información gerencial que integra muchas de las prácticas de los negocios asociados con las operaciones de producción y gestión.

-F-

Framework: Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

-G-

Generador de informes o reportes: Herramienta que permite generar informes, también conocidos como informes a partir de datos primarios.

GALBA: Guardián del Alba.

-H-

HTML (HyperText Markup Language, lenguaje de marcas hipertextuales): Lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

-I-

Informe o reporte: Documento que contiene información personalizada y útil para el destinatario del mismo.

-M-

Multiplataforma: Es un término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

Multiproceso: Modalidad operativa de un sistema informático que permite la ejecución simultánea de varios programas compartiendo la memoria central y las unidades periféricas.

Multihilo: un hilo, es un flujo de control secuencial dentro de un programa, no puede correr por sí mismo, se ejecuta dentro de un programa. Se pueden programar múltiples hilos de ejecución para que corran

simultáneamente en el mismo programa. La utilidad de la programación multihilos resulta evidente. Por ejemplo, un navegador Web puede descargar un archivo de un sitio, y acceder a otro sitio al mismo tiempo. Si el navegador puede realizar simultáneamente dos tareas, no tendrá que esperar hasta que el archivo haya terminado de descargarse para poder navegar a otro sitio.

-O-

ODBC (Open DataBase Connectivity): es un estándar de acceso a bases de datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos, ODBC logra esto al insertar una capa intermedia llamada manejador de bases de datos, entre la aplicación y el DBMS.

-P-

PDF (Portable Document Format, Formato de Documento Portátil): Formato de almacenamiento de documentos multiplataforma (Microsoft Windows, Unix, Mac) desarrollado por la empresa Adobe System. Especialmente ideado para documentos susceptibles de ser impresos.

-S-

SDK: Un kit de desarrollo de software o SDK (siglas en inglés de software development kit) es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, entre otros. Los SDK frecuentemente incluyen, también, códigos de ejemplo y notas técnicas de soporte u otra documentación de soporte para ayudar a clarificar ciertos puntos del material de referencia primario.

SCADA: Supervisión Control y Adquisición de Datos.

-W-

Widget: es una aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de widgets o Widget Engine. Entre sus objetivos están los de dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.