INSTITUTO SUPERIOR MINERO METALÚRGICO DE MOA

"DR. ANTONIO NÚÑEZ JIMÉNEZ"

FACULTAD DE METALURGIA Y ELECTROMECÁNICA

Generador de Reportes para el Sistema de Supervisión y Control de Procesos EROS

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA

AUTOR

Wilker Juan Urgellés Cobos

TUTOR

Lic. Isidro Manuel Corría Ramírez

Moa

Julio 2007

INSTITUTO SUPERIOR MINERO METALÚRGICO DE MOA

"DR. ANTONIO NÚÑEZ JIMÉNEZ"

FACULTAD DE METALURGIA Y ELECTROMECÁNICA

Generador de Reportes para el Sistema de Supervisión y Control de Procesos EROS

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA

AUTOR

Wilker Juan Urgellés Cobos

TUTOR

Lic. Isidro Manuel Corría Ramírez

Moa

Julio 2007

Dedicatoria

No existen palabras para describir el amor, el cariño, el apoyo y la confianza que he recibido en este largo camino por mi superación integral de parte de mis padres, a ellos dedico este trabajo.

Wilker Juan Urgellés Cobos.

Agradecimientos

A mi familia, por tanto amor, cariño y apoyo en todo momento.

A mis amigos por la preocupación que siempre han tenido por mí.

A los tutores que he tenido, a mis compañeros de grupo y de la residencia estudiantil.

A mis profesores y a los del departamento de informática.

A todos aquellos que de una forma u otra han contribuido con mi formación como futuro profesional y a la realización de este trabajo.

Muchas gracias,

Wilker Juan Urgellés Cobos.

Resumen

El presente trabajo expone detalles de la elaboración de un Generador de Reportes para el Sistema de Supervisión y Control de Procesos EROS, que no contaba con este servicio para sus usuarios. Se incluye una panorámica del estado del arte de la materia en el mundo y el estudio de la factibilidad del proyecto. Quedan expuestos los pormenores del diseño e implementación: modelos, algoritmos, diseño, clases y herramientas empleadas.

Abstract

The present work showes details of the elaboration of a Generator of Reports for the Supervision and Control System of Processes EROS which didn't have, up to now, this service for its users. A panoramic of the state of the art of the matter in the world is included as well as the study of the feasibility of the project. The details of the design and implementation are shown: models, algorithms, design, classes and used tools.

Índice

Lista de Tablas

Lista de Figuras

Introducción

El constante desarrollo del Sistema de Supervisión y Control de Procesos EROS (SCADA por sus siglas en inglés) y sus múltiples facilidades para operar y dirigir cualquier industria, trabajando con diversos sistemas de colección de datos (autómatas programables, estaciones de adquisición de datos y reguladores autónomos), como elemento único o formando parte de una red industrial que por su eficacia avalada en más de 66 aplicaciones en plantas industriales dentro y fuera del país en el lapso de 10 años pone a los usuarios en la situación de buscar una vía que le facilite la creación de reportes según sus necesidades, con los datos que son captados por las estaciones de medición.

La creación de reportes es medular para cualquier usuario del sistema en una industria, pues este tiene ante sí un océano de variables disponibles, de las cuales no todas le son útiles dependiendo de la actividad que realice. El problema puede verse en ámbitos más estrechos, especialmente, sí se analiza la generación de reportes en determinadas industrias o departamentos por separados.

Uno de los mayores problemas que tiene la generación de reportes es el hecho de que en algunos casos las variables pueden estar distribuidas en más de una estación de medición como parte de una red industrial. Otro problema, es el hecho de que el usuario interesado en recibir un reporte no tiene el modo de configurarlo de acuerdo con sus necesidades, incluyendo tablas y gráficos para que este se genere de forma periódica o progresiva, además de que en algunas ocasiones se imprima o se publique en Web de modo automático.

Según lo antes expuesto se puede resumir **la situación problémica** en:

- Necesidad de los usuarios de obtener reportes.
- Dispersión de las variables distribuidas en las estaciones de medición.
- Configuración de reportes en correspondencia con las necesidades de los usuarios.

Con este trabajo se pretende dar soluciones a las situaciones anteriormente expuestas, definiéndose como problema**, la no existencia de un generador de reportes para los usuarios del Sistema de Supervisión y Control de Procesos EROS**.

Entonces se impone la necesidad de crear un Generador de Reportes que localice las variables y las disponga en función de operadores, ingenieros, especialistas, supervisores y directivos, los cuales seleccionarán las que necesiten para construir un modelo de reporte según sus necesidades, facilitando así todo el proceso posterior de generación.

Los reportes en general adquieren gran importancia en las industrias cuando de tomar decisiones se trata, mientras más disponibles estén y más confiables sean, más acertadas serán las decisiones que se tomen a partir de ellos, para alcanzar altos grados de eficiencia en el proceso de generación. Los usuarios o grupos deben dedicarle más tiempo a la forma en que se mostrará la información final en los reportes, partiendo de los datos captados de las estaciones de medición.

Para analizar los **antecedentes de los reportes en el EROS** debemos remontarnos a la historia de este SCADA. En 1973 el Grupo SAD-PT de la Empresa Comandante René Ramos Latourt (ECRRL) introduce la Computación en la supervisión de los Procesos Tecnológicos, acoplando una minicomputadora cubana CID201B a un equipo de adquisición de datos en la Planta de Hornos de Reducción, emitiéndose reportes de producción mediante teletipos. Ya para 1983, se comienza a trabajar con microprocesadores (I8085) realizándose los primeros diseños de las Estaciones Locales de Medición y la primera versión del Sistema de Pesaje Electrónico. En 1993 a propuesta del grupo SAD-PT de la ECRRL se crea en la Unión del Níquel un grupo de trabajo para la aplicación de los medios superiores de cómputo en la tarea de supervisión de procesos tecnológicos y a finales de 1993 sale a la luz la primera versión del Sistema de Supervisión EROS (sobre el sistema operativo DOS) y se instala en la Planta de Lixiviación y Lavado de la Empresa Comandante Ernesto Che Guevara (ECG) comunicándose con equipos de adquisición de datos fabricados por el Grupo SAD-PT. En 1995 se termina la versión del Sistema de Pesaje Electrónico, que se encuentra hoy contabilizando el mineral de la ECG. Se continuaron desarrollando diferentes versiones del EROS los cuales contenían un generador de reporte interno que emitían los resúmenes en forma de notas, ya en 1996 se transfiere al sistema operativo Windows 95, pasando por varias actualizaciones para estar disponible en Windows 98 y NT, en 1998 el Grupo pasa a la Empresa de Servicios de Electrónica, Informática y Comunicaciones del Níquel (SerCoNi) y

se crea un Sistema de Generación de Reportes constituyendo un paquete aparte y una licencia específica, excluyéndose así el que existía del ejecutable del EROS.

El EROS como SCADA actualmente no cuenta con un sistema para la generación de reportes, los que existían anteriormente no cumplían las necesidades actuales de los usuarios.

El Objeto de Estudio de este trabajo de diploma en cuestión es la creación de reportes para los usuarios del Sistema Supervisión y Control de Procesos EROS mediante un Generador de Reportes.

El **Campo de Acción** se enmarca en la implementación a modo de prueba de un Generador de Reportes mediante una aplicación de software.

Como **Hipótesis** se parte de la idea de que si se desarrolla un Generador de Reporte, basado en la potencia de edición de tablas y gráficos que posee el Microsoft Excel combinado con la posibilidad de insertar en celdas seleccionadas los valores que se necesiten de las estaciones de medición del EROS y el lenguaje de programación DELPHI, es posible lograr una aplicación de sencilla configuración que de forma rápida y fiable brinde reportes en los que pueden aparecer valores puntuales, cálculos estadísticos en un determinado período (hora, turno o día), además de tablas, gráficos e imágenes.

El **Objetivo General** del trabajo de diploma es analizar, diseñar e implementar a modo de prueba un Generador de Reportes para el Sistema de Supervisión y Control de Procesos Industriales EROS.

Se plantean los siguientes **objetivos específicos**:

- Analizar el Generador de Reportes.
- Diseñar el Generador de Reportes.
- Implementar a modo de prueba el Generador Reportes.
- Analizar la factibilidad del proyecto.

Para cumplir con estos objetivos y resolver la situación problémica planteada, se proponen las siguientes **tareas**:

- Revisión Bibliográfica acerca de los Generadores de Reportes en SCADAS.

- Análisis, Diseño e Implementación a modo de prueba del Generador de Reportes.
- Estudio de la factibilidad del proyecto, centrado en estimaciones de esfuerzo humano, tiempo de desarrollo para su ejecución y costos.

El presente trabajo de diploma tiene como **Aportes** la propuesta de un Generador de Reportes que les brinda a los usuarios del Sistema de Supervisión y Control de Procesos EROS una aplicación de sencilla configuración y fiable que les permita localizar las variables para disponerlas en función de crear reportes según sus necesidades en los que pueden aparecer valores puntuales, cálculos estadísticos en un determinado período (hora, turno o día), además de tablas, gráficos e imágenes, aumentando el nivel de información del proceso y grado de eficiencia en la toma de decisiones de los operadores, ingenieros, supervisores y directivos de la industria automatizada.

El presente trabajo costa de introducción, tres capítulos, conclusiones, recomendaciones, bibliografía, glosario de términos y anexos.

En el **Capítulo 1** se profundiza en los fundamentos teóricos de la investigación, abordando aspectos necesarios para el entendimiento del tema, que se encuentran relacionados con el dominio del problema y que ayudarán al entendimiento del negocio y de la propuesta de solución. Trata la situación de las tecnologías a utilizar en el desarrollo de la propuesta, y se explican los conceptos principales que se van a tratar a lo largo del trabajo. Establece las herramientas y metodologías a utilizar.

El **Capítulo 2** describe el negocio a través de un Modelo de Dominio, y a partir de este se comienza a hacer el diseño del sistema a desarrollar, se definen sus funcionalidades y se describen, utilizando las herramientas de modelación, elaborando los diagramas de clases, secuencia de datos, despliegue y de componentes.

En el **Capítulo 3** se siguen una serie de pasos encaminados a conformar un estudio de factibilidad del sistema. Se reflejan los cálculos intermedios, así como las tablas con datos de la aplicación y resultados finales. Se analizan los beneficios tangibles e intangibles y los costos del desarrollo de la propuesta.

1

Capítulo

Fundamentos Teóricos.

Durante el desarrollo de este capítulo, se hace referencia a una serie de conceptos, cuya terminología formará parte inseparable de usuarios y desarrolladores. Hacemos referencia a algunos generadores de reportes que se han realizado a nivel mundial para los sistemas de supervisión y control de procesos que por sus características son de usos específicos por tener cada uno su propio modo para adquirir los datos respondiendo a las exigencias de su productor.

1.1 Estado del Arte.

Un SCADA viene de las siglas de "Supervisory Control And Data Adquisition", es decir: adquisición de datos y control de supervisión. Se trata de una aplicación de software, especialmente diseñada para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo (controladores autónomos y autómatas programables) y controlando el proceso de forma automática desde la pantalla del ordenador. Además, brinda toda la información que se genera en el proceso productivo a diversos usuarios dentro de la industria.

En este tipo de sistemas usualmente existe un ordenador, que efectúa tareas de supervisión y gestión de alarmas, así como tratamiento de datos y control de procesos. La comunicación se realiza mediante buses especiales o redes LAN. Todo esto se ejecuta normalmente en tiempo real, y están diseñados para dar al operador de planta la posibilidad de supervisar y controlar dichos procesos.

La mayoría de los SCADAS tienen un sistema para generar informes y reportes a los usuarios de los datos que se obtienen de las estaciones de medición.

Entre los SCADAS con sistemas para la generación de reportes encontramos:

1. **Paradym-31**, provee un ambiente gráfico de programación compatible con MS Windows, que permite construir programas de control en tiempo real, tales como los tradicionales Controladores Lógicos Programables (PLCs). Utilizado en conjunto con el módulo de control ADAM-5510, este software es capaz de brindar una solución completa de automatización. Por ser compatible con la norma IEC 1131-3 se reduce significativamente el costo de programación y entrenamiento. El usuario puede construir sus propias funciones lógicas y generar reportes automatizados especiales. Su proveedor es Advantch[1].
2. **WizFactory**, solución completa para información y automatización, combina el control discreto y el continuo con SCADA e Internet. Entre sus componentes se encuentra Wizcon para Windows e Internet, que es una herramienta poderosa para canalizar información en tiempo real e histórica de la planta. Provee funciones completas de SCADA y HMI, las cuales se pueden también visualizar a través de la red de redes mediante un navegador convencional. Otro componente es WizPLC, solución que permite emular en cualquier PC el comportamiento lógico de un PLC. Por su parte, WizDCS emula el comportamiento de un DCS en una PC. Finalmente, WizReport facilita la generación de reportes basados en los datos producidos por todas las aplicaciones anteriores. Su proveedor es eMation[2].
3. **Genesis32**, desde un inicio se diseñó para beneficiarse de las nuevas tecnologías orientadas a objetos, tales como la arquitectura DNA de Microsoft 95/98/2000, que incluye VBA, COM, DCOM y ActiveX. Al mismo tiempo, en el corazón del producto se encuentra ubicada la tecnología moderna de OPC. Es un producto de rendimiento óptimo cuando se utiliza para construir aplicaciones de control y automatización que requieran visualización, control supervisorio, adquisición de datos, sistemas avanzados de alarmas, SPC/SQC (control estadístico de procesos y calidad), sistemas de reportes y administración de recetas en procesos de bache. Las aplicaciones desarrolladas con este producto se integran fácilmente con otros sistemas de nivel superior, tales como

MES (sistemas de ejecución para manufactura) y MRP (planeación de materiales). Su proveedor es Iconics[3].

4. **Lookout 4.5**, Proporciona Control ActiveX para aplicaciones industriales, y los usuarios pueden aprovechar cualquier producto con control ActiveX disponible ya sea de NI o de terceros para construir dichas aplicaciones. La más reciente versión del software HMI/SCADA orientado a objetos y de fácil uso es un contenedor ActiveX para integrar y controlar objetos, y desarrollar las aplicaciones de manera sencilla y rápida. Otra importante característica es su integración plena con las funciones de Internet, como es la creación de reportes HTML, envío de correos electrónicos y exportar algunos procesos a través de la Web para no solamente monitorear, sino controlar algunos procesos en forma remota. Su proveedor es National Instruments[4].
5. **WinCC HMI Ver. 5.0**, para integrar software en la manufactura se requiere usar normas abiertas que puedan enlazar fuertemente la información del piso de la planta y el sistema de negocios a través de ella. WinCC HMI, software de 32 bits integrado completamente con Microsoft Windows NT, combina las características estándar (gráficos, alarmas, administración de recetas, etcétera) con otras avanzadas (reportes, referencias entre proyectos, diagnóstico de proceso, soporte multilingüe y redundancia completa). Además, mejora su funcionalidad mediante la integración de bases de datos con MES/ERP, Internet y tecnologías de cliente delgado. Su proveedor es Rockwell Automation[5].
6. **Web Control Center (webCC)**, permite que una gran variedad de información desde varias áreas de la planta se despliegue simultáneamente en una sola interfase de usuario. Así, se facilita el control corporativo y la operación y monitoreo de las instalaciones de proceso y plantas de producción. Mediante un navegador de Internet basado en Java se tiene acceso simultáneo a diversos sistemas de control central, tales como SiiX-IS, WinCC, Sicalis PMC y Simatic PCS 7. Bases de datos y videos se integran desde la red y pueden enviarse correos electrónicos en cualquier momento. Los datos maestros de sistemas SAP y otros pueden consultarse desde cualquier parte del mundo haciendo realidad el concepto de compañía virtual. Su proveedor es Rockwell Automation[5].

7. **Aimax**, muy robusto y poderoso en la categoría de MMI, este software opera en la plataforma Microsoft Windows y es capaz de almacenar e integrar datos de múltiples fuentes, gracias a la disponibilidad de interfaces para una amplia gama de, PLCs, controladores y dispositivos de entrada/salida. Provee diversas funciones, tales como: adquisición de datos, alarmas, gráficas, archivos históricos y reportes. Es muy fácil de configurar utilizando interfases estándar de Microsoft Windows basadas en Win32 API y arquitectura de componentes (COM, MFC, OPC). Se cuenta con una librería de cientos de símbolos preconstruidos que facilitan la elaboración de gráficos dinámicos. Conserva compatibilidad plena con la mayoría de los fabricantes de PLCs (Allen-Bradley, Modicon y Siemens, entre otros) y cuenta con una base de datos relacional propietaria que le proporciona un desempeño mejorado y una gran flexibilidad para el manejo de los datos. Su proveedor es TA-Engineering Products[6].

Los Sistemas de Reportes de los SCADAS mencionados anteriormente, en su mayoría realizan reportes rígidos en su formato, es decir, que los usuarios no pueden modificar sus estructuras. Están diseñados para uso específico de los SCADA de los que se sirven como clientes y para los cuales fueron implementados, por lo que no se adaptan a las características del Sistema de Supervisión y Control de Procesos EROS que para ser competitivo en el Mercado debe también tener un generador de reportes.

Para analizar el estado del arte de los Reportes en el EROS debemos remontarnos a la historia de este SCADA. En 1973 el Grupo SAD-PT de la Empresa Comandante René Ramos Latourt (ECRRL) introduce la Computación en la supervisión de los Procesos Tecnológicos, acoplando una minicomputadora cubana CID201B a un equipo de adquisición de datos en la Planta de Hornos de Reducción, emitiéndose reportes de producción mediante teletipos. Ya para 1983, se comienza a trabajar con microprocesadores (I8085) realizándose los primeros diseños de las Estaciones Locales de Medición y la primera versión del Sistema de Pesaje Electrónico. En 1993 a propuesta del grupo SAD-PT de la ECRRL se crea en la Unión del Níquel un grupo de trabajo para la aplicación de los medios superiores de cómputo en la tarea de supervisión de procesos tecnológicos y a finales de 1993 sale a la luz la primera versión del Sistema de Supervisión EROS (sobre el sistema operativo DOS) y se instala en la Planta de Lixiviación y Lavado de la Empresa Comandante Ernesto Che Guevara (ECG)

comunicándose con equipos de adquisición de datos fabricados por el Grupo SAD-PT. En 1995 se termina la versión del Sistema de Pesaje Electrónico que se encuentra hoy contabilizando el mineral de la ECG. Se continuaron desarrollando diferentes versiones del EROS los cuales contenían un generador de reporte interno que emitía los resúmenes en forma de notas, ya en 1996 se transfiere al sistema operativo Windows 95, pasando por varias actualizaciones para estar disponible en Windows 98 y NT, en 1998 el Grupo pasa a la Empresa de Servicios de Electrónica, Informática y Comunicaciones del Níquel (SerCoNi) y se crea un Sistema de Generación de Reportes constituyendo un paquete aparte y una licencia específica, excluyéndose así el que existía del ejecutable del EROS.

El EROS como SCADA actualmente no cuenta con un Sistema para la generación de reportes ya que los que existían anteriormente no cumplían las necesidades actuales de los usuarios.

Entonces, si utilizamos las potencialidades de edición de tablas y gráficos que posee el Microsoft Excel combinado con la posibilidad de insertar en celdas seleccionadas los valores que se necesiten de las estaciones de medición del EROS podemos obtener un Generador de Reportes que nos daría la posibilidad de que los usuarios configuren sus reportes de forma sencilla y con mucha flexibilidad, además de poder optar con la posibilidad de exportarlos automáticamente a Web. Claro, debemos pensar primeramente, en facilitarles a los usuarios una programación horaria bastante variable para que puedan configurar sus reportes teniendo en cuenta el tiempo de generación que deseen (mensual, quincenal, decenal, semanal, por día de la semana, por día del mes, o por hora), además de garantizar que tengan acceso a las variables que necesiten de las estaciones de medición para tomar sus valores puntuales, cálculos estadísticos en un determinado período (hora, turno o día).

1.2 Fundamento de la metodología a utilizar y lenguaje de programación.

La calidad en el desarrollo y mantenimiento del software se ha convertido hoy día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales dependen de los sistemas informáticos para su buen funcionamiento. En los últimos años se han publicado diversos estudios y estándares en los que se exponen los principios que se deben seguir para la mejora de los procesos de software.

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas Informáticos. Por ello, escoger la metodología que va a guiar el proceso de desarrollo del sistema es un paso tan importante.

Para controlar y planificar la propuesta que presenta este trabajo, se decidió utilizar como metodología el Proceso Unificado de Modelado (RUP), por sus características y las facilidades que aporta a todo el proceso. Y teniendo en cuenta de que viene acompañado de una herramienta muy buena que soporta cada uno de los procesos que se necesitó la "suite" Rational Rose Enterprise Edition 2000 y el diseño sobre UML.

1.2.1 UML.

El UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. UML proporciona una forma estándar de escribir los planos de un sistema, cubriendo las cosas conceptuales como: procesos del negocio y funciones del sistema; las cosas concretas como: las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables.

1.2.2 RUP.

Se hizo uso de las herramientas de la metodología RUP (Rational Unified Process) para facilitar el desarrollo del sistema.

El Proceso Unificado es un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, tipos de organizaciones, niveles de actitud y tamaños de proyecto. Está basado en componentes, que quiere decir que el sistema software en construcción está formado por componentes de software interconectados a través de interfaces bien definidas. Utiliza el Lenguaje Unificado de Modelado (Unified Modeling Languaje, UML) para preparar todos los esquemas de un

sistema software. Garantiza la elaboración de todas las fases de un producto de software orientado a objetos.

UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

1.2.3 Características del Proceso Unificado.

Los verdaderos aspectos definitorios del Proceso Unificado, y que lo convierten en único, se resumen en tres frases claves - dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

- Dirigido por los casos de uso: Teniendo en cuenta que la razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como el conjunto de acciones que debe realizar un sistema para dar un resultado de valor a un determinado usuario y los utiliza tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígase diseño, implementación y prueba.
- Estar centrado en la arquitectura: La arquitectura es una vista del diseño completo con las características más importantes, dejando a un lado los detalles. Esta no solo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red; con los que debe coexistir el sistema. En otras palabras, la arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.
- Ser iterativo e incremental: La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos mini-proyectos se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos.

1.2.4 Rational Rose.

Es una herramienta para "modelado visual", que forma parte de un conjunto más amplio de herramientas que juntas cubren todo el ciclo de vida del desarrollo de software. Permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP) e incluye un conjunto de herramientas de ingeniería inversa y generación de código que allanan el camino hasta el producto final.

1.2.5 Borland Delphi.

Borland Delphi fue el entorno de Programación Visual Orientado a Objetos que se optó para la creación de nuestro Generador de Reportes porque forma parte de la directiva de la Empresa SerCoNi trabajar con el mismo en la creación de sus aplicaciones como el Sistema de Supervisión y Control de Procesos EROS, además porque permite el desarrollo rápido de aplicaciones (RAD) de propósito general, incluyendo aplicaciones cliente/servidor. Desarrollo de bases de datos multinivel dimensionable, auténtica capacidad de reutilización orientada a objetos y compilador de código original de alto rendimiento, incluye numerosas mejoras, incluyendo soporte para Bases de Datos distribuidas, creación de componentes ActiveX, DLLs de componentes, así como algunas novedades en el lenguaje de programación.

Delphi tiene las siguientes características:

- Rendimiento – con el mejor y más rápido compilador del mundo.
- Empresa e Internet - Soluciones cliente y servidor.
- Desarrollo de aplicaciones rápidas (RAD).
- Reusabilidad de componentes, un verdadero entorno orientado a objetos.
- Manejo de Bases de Datos escalables.
- Arquitectura multinivel abierta y dimensionable.
- Fábrica de componentes.
- Diseminación de información de base de datos en la Web a una gran velocidad.
- Facilidad y productividad mejoradas.

Permite crear aplicaciones, de alta velocidad y alto rendimiento con controladores nativos a sistemas anteriores de datos a los que antes no podía acceder. La arquitectura abierta de Delphi y su compatibilidad con DLL de sistema nativo constituyen la base de esta alta productividad en el desarrollo rápido de aplicaciones para Internet.

Los controladores nativos ofrecen mayor rendimiento y más potencia que la solución del mínimo común denominador, ODBC. El Driver Development Kit (DDK) permite a las empresas crear controladores nativos para el Borland Database Engine (Motor de base de datos) y complementar la nueva tecnología Remote DataSet. Los controladores nativos creados con el DDK pueden usarse en todos los productos de la familia Borland incluidos IntraBuilder, C++, Delphi y Open Jbuilder.

Incluye plantillas estándar de código para sentencias como If, FOR, WHILE y CASE para que la sintaxis de programación siempre sea correcta es necesario iniciar en el editor una sentencia de código y Delphi la completará, reduciendo así las posibilidades de error y acelerando el proceso de desarrollo. Delphi también permite al desarrollador añadir sus propias plantillas de código para adaptar el entorno de desarrollo a los estándares y criterios corporativos.

Simplifica la reutilización de componentes, gracias a la Creación de componentes visuales. Los desarrolladores pueden crear fácil e instantáneamente componentes combinados con su código asociado y colocar el nuevo componente en la página de la paleta de Delphi. Posteriormente, el desarrollador podrá reutilizar esos objetos en el mismo proyecto o en otro, lo que facilita y agiliza el proceso de desarrollo.

Permite al desarrollador depurar DLL en el entorno Delphi. No es preciso disponer de un costoso programa de depuración independiente para poder crear aplicaciones complejas con DLL. Simplemente seleccionando la aplicación anfitriona y estableciendo un punto de ruptura en la DLL, el desarrollador puede agilizar mucho la creación y depuración de archivos DLL empleados en los WebServers y otras herramientas.

Es compatible con una gran variedad de motores (ENGINES) de bases de datos gracias a una API abierta, por lo que todos los conjuntos de resultados de Database Engine funcionan perfectamente con los potentes controles de Delphi enlazados a bases de datos. Usa una arquitectura abierta para poder ser compatible con numerosos motores ligeros de bases de

datos. El desarrollador controla completamente los servicios esenciales de bases de datos (como ubicación de datos en la memoria caché, recuperación, intercalación de idiomas, acceso heterogéneo, compatibilidad de cursores genéricos) necesarios para una aplicación determinada.

Integra perfectamente el Modelo de Objeto Común, COM (Common Object Model) de Microsoft en su entorno de desarrollo rápido de aplicaciones para que los desarrolladores puedan crear, fácil y rápidamente, objetos de empresa reutilizables e interactuables mediante lenguaje. Delphi usa los objetos COM en un entorno multinivel para proporcionar soluciones empresariales integradas y reutilizables.

También proporciona una forma de herencia múltiple: a partir de ahora, es posible tener una clase que deriva de dos o más clases. La herencia múltiple tal y como se implementa en algunos lenguajes, como C++.

2

Capítulo

Análisis, Diseño e Implementación.

En el presente capítulo se describen los procesos del negocio que tiene que ver con el objeto de estudio y teniendo en cuenta el bajo nivel de estructuración, se desarrollará un modelo de dominio, el cual permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Se enumeran los requisitos funcionales y no funcionales que debe tener el sistema que se propone, lo que permite hacer una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema, y las secuencias de acciones con las que interactúan. Se describen también los diagramas de clases, secuencia, despliegue y de componentes.

2.1 Modelo de Dominio.

Un Modelo de Dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las ¨ cosas ¨ que existen o los eventos que suceden en el entorno en el que trabaja el sistema. [7].

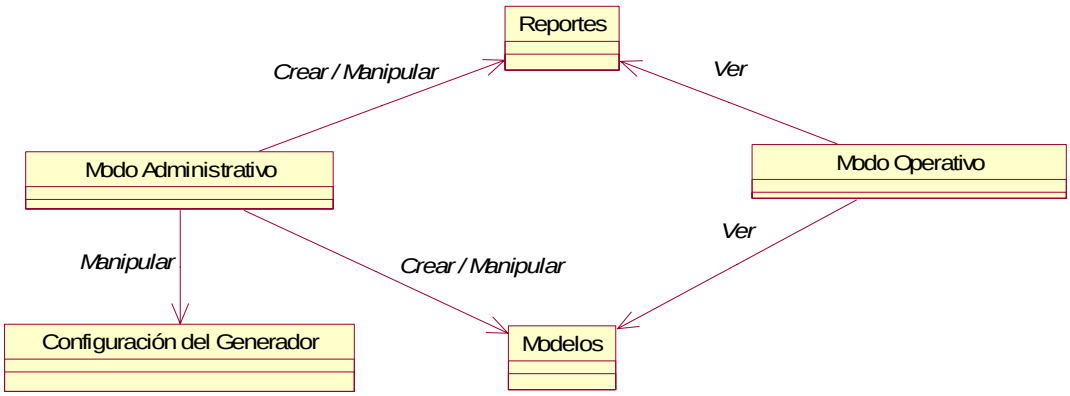

Figura 2.1 Modelo de Dominio.

2.2 Requerimientos Funcionales del Sistema.

Los requerimientos funcionales definen las responsabilidades del sistema, o sea, las funciones que éste será capaz de realizar. Posteriormente estos requisitos son modelados a través del diagrama de casos de uso del sistema. A continuación lo enumeraremos:

1. Validar Modo. (Modo Administrativo o Modo Operativo).
2. Configurar Generador de Reportes.
3. Crear Modelo de Reporte.
4. Modificar Modelo de Reporte.
5. Eliminar Modelo de Reporte.
6. Ver Modelo de Reporte.
7. Crear Reporte.
8. Eliminar Reporte.
9. Ver Reporte.
10. Exportar Reporte Generado a Web.
11. Imprimir Reporte Generado.
12. Generar Reporte Configurado de forma Manual.

2.3 Requisitos no Funcionales del Sistema.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

2.3.1 Requerimientos de Apariencia o Interfaz Externa.

La interfaz debe ser agradable para conseguir la confianza de los usuarios en la utilización del generador de reportes, con colores claros preferiblemente. Se debe tener en cuenta algunos elementos de diseño como gráficos de encabezamiento, estilos y formatos de texto, estilo y paletas de color de los gráficos y colores o patrones del fondo.

2.3.2 Requerimientos de Rendimiento.

Por la importancia que el proceso de Generación de Reportes tiene, esta aplicación debe tener un rendimiento óptimo. La aplicación propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la programación horaria de los reportes.

2.3.3 Requerimientos de Disponibilidad.

Se les garantizará a los usuarios del Generador el acceso total a los reportes, logrando la mayor rapidez en el acceso a los datos deseados en un momento dado.

2.3.4 Requerimientos de Software y de Hardware.

Para lograr un buen funcionamiento de este sistema se necesita tener instalado el Microsoft Excel en una de sus versiones y una computadora con Microprocesador Pentium II o superior con más de 600 Mhz y espacio de disco duro dependiendo de la cantidad de reportes generados que se quiera guardar, se recomienda al menos 1 Gb de espacio libre. En caso de que se vaya a trabajar conectado a una Red industrial contar con una tarjeta de red.

2.4 Diagrama de Caso de Uso del Sistema.

Utilizando las facilidades que brinda el UML, se representarán los requisitos funcionales del sistema mediante un diagrama de casos de uso. Para ello se definirán primeramente cuáles son los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades.

Un caso de uso es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor.[8] Un actor no es parte del sistema, representa un rol que es jugado por una persona, un dispositivo hardware o incluso otro sistema al interactuar con nuestro sistema. Puede intercambiar información o puede ser un recipiente pasivo de información.

Tabla 2.1 Justificación de Actores.

ACTORES	JUSTIFICACIÓN
Operador	Es usuario del sistema y puede: - Ver Modelos de Reportes. - Ver Reportes. - Exportar Reportes Configurados a Web. - Imprimir Reportes Generados. - Generar Manualmente Reportes Configurados.
Administrador	Es usuario y a su vez administrador del sistema y puede hacer todas las cosas del Operador además de: - Crear Modelos de Reportes. - Crear Reportes. - Eliminar Modelos de Reportes. - Eliminar Reportes. - Configurar el Sistema.

A continuación se presentan los casos de uso determinados para satisfacer los requerimientos funcionales del sistema:

Nombre del caso de uso	Validar Modo.
Actores	Operador (inicia).
Propósito	Validar el acceso al Modo Administrativo.
Resumen. El Operador coloca la contraseña para Abrir el Modo Administrativo.	
Referencias	R1
Precondiciones	El Generador debe estar en Modo Operativo.
Poscondiciones	Queda Abierto el Modo Administrativo.

Nombre del caso de uso	Configurar Generador.
Actores	Administrador (inicia).
Propósito	Configurar la Administración del Generador de Reportes.
Resumen. El Administrador debe definir la contraseña y el tiempo de duración para el Modo Administrativo y si se publicarán o no automáticamente los reportes en formato Web.	
Referencias	R2 Validar Modo<<include>>
Precondiciones	El Generador debe estar en Modo Administrativo.
Poscondiciones	Queda configurada la Administración del Generador de Reportes.

Nombre del caso de uso	Crear Reporte.
Actores	Administrador (inicia).
Propósito	Crear un reporte.
Resumen. El Administrador selecciona el modelo configurado con anterioridad para el nuevo reporte y de este modo es creado.	
Referencias	R7 Validar Modo<<include>>
Precondiciones	El Generador debe estar en Modo Administrativo.
Poscondiciones	Queda creado el reporte.

Nombre del caso de uso	Exportar Reporte Web.
Actores	Operador o Administrador (inicia).
Propósito	Exportar un reporte generado a Web.
Resumen. El Operador o el Administrador del Generador seleccionan el reporte generado que desean exportar a Web y el reporte es exportado.	
Referencias	R10
Precondiciones	
Poscondiciones	Queda exportado a Web el reporte generado seleccionado.

El resto de las descripciones de los Casos de Usos se encuentran en el Anexo 1.

El diagrama donde se muestra la relación existente entre los actores y los casos de uso desde un punto de vista estático se representa a continuación:

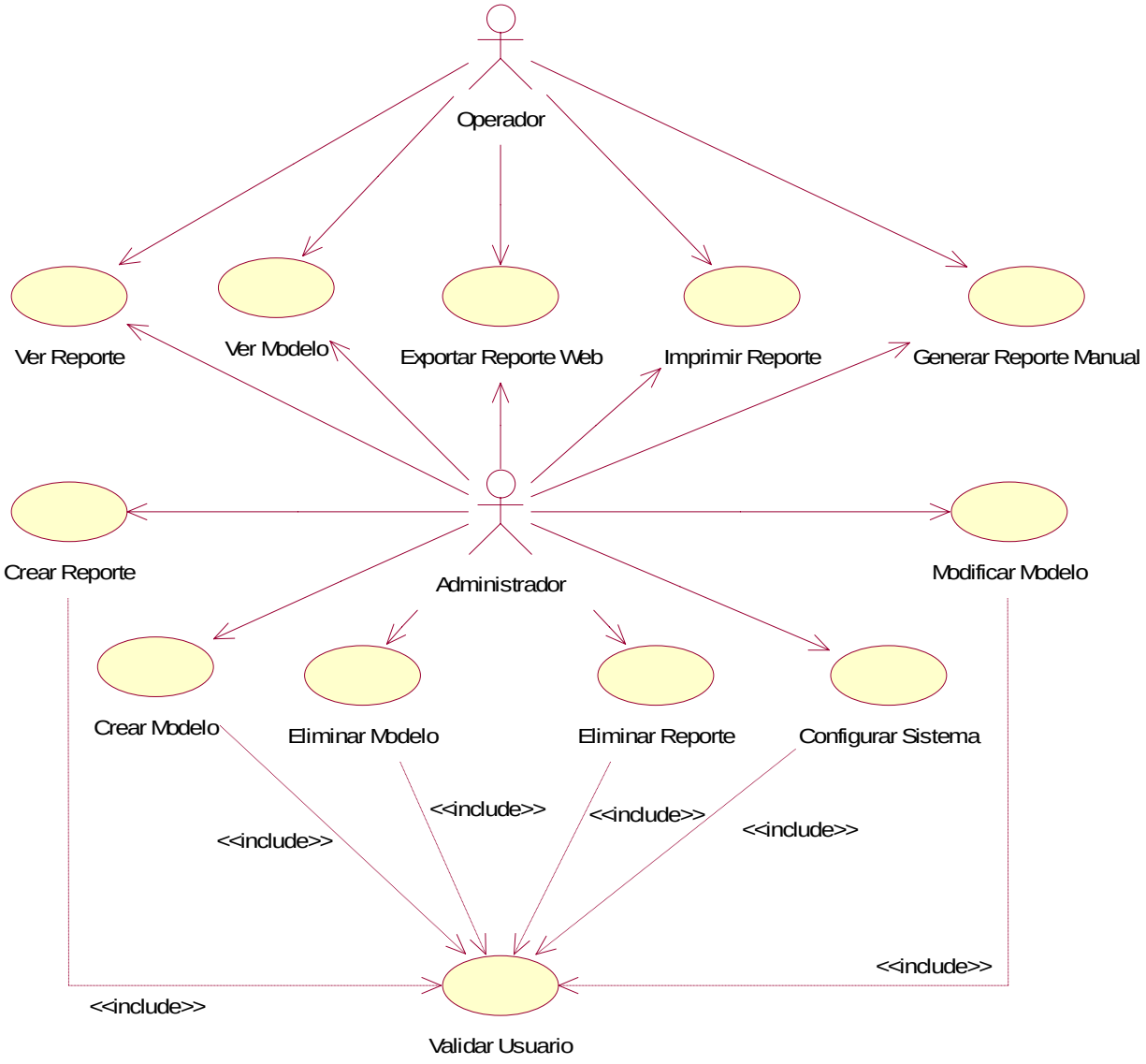

Figura 2.2 Diagrama de los Casos de Usos del Sistema.

2.5 Diagramas de Interacción del Diseño.

Los diagramas de interacción tratan la vista dinámica del sistema. Muestra la interacción, consistente en un conjunto de objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos.

Ahora mostraremos los diagramas de secuencia de los casos de usos:

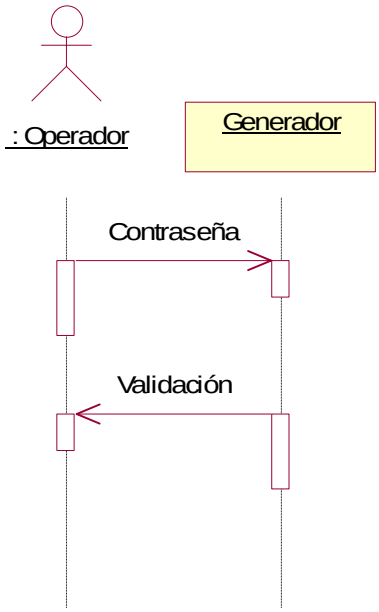

Figura 2.3 Diagrama de Secuencia del Caso de Uso Validar Modo.

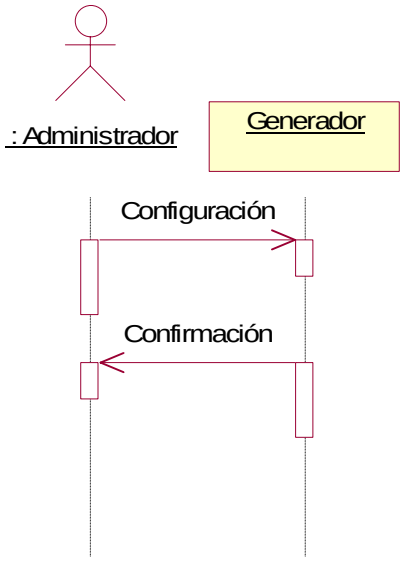

Figura 2.4 Diagrama de Secuencia del Caso de Uso Configurar Generador.

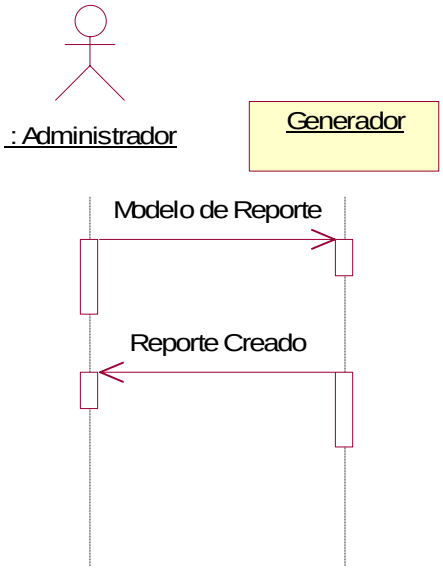

Figura 2.5 Diagrama de Secuencia del Caso de Uso Crear Reporte.

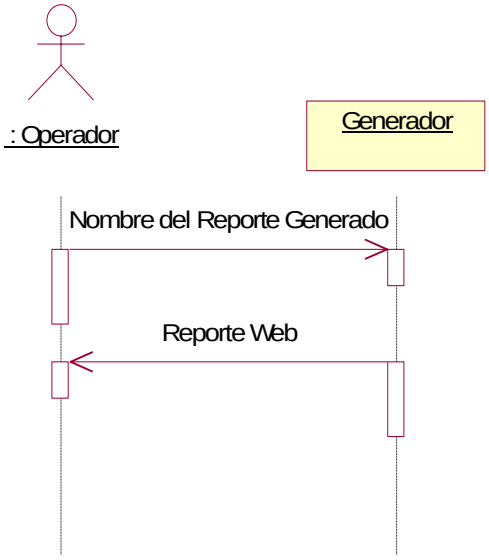

Figura 2.6 Diagrama de Secuencia del Caso de Uso Exportar Reporte Web.

El resto de los Diagramas de Secuencia de los Casos de Usos se encuentran en el Anexo 2.

2.6 Diagrama de clases del Diseño.

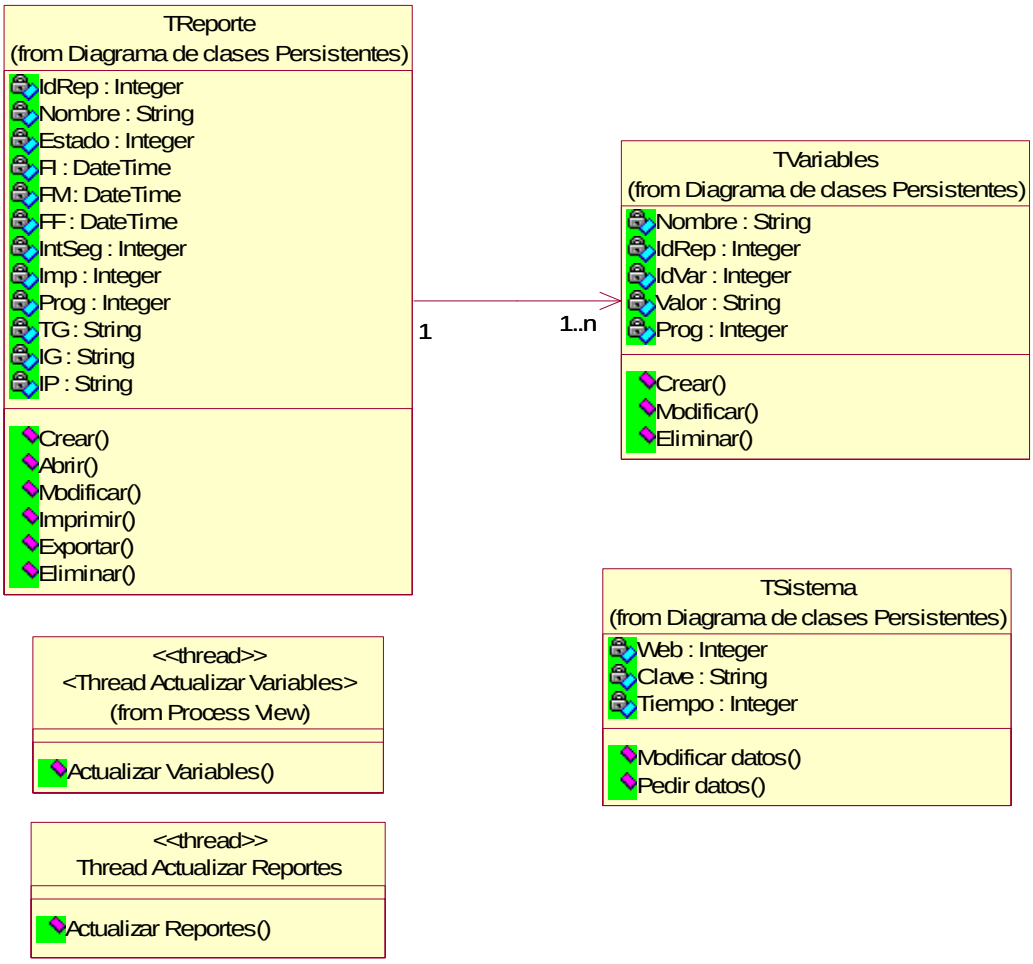

Figura 2.7 Diagrama de Clases.

2.6.1 Descripción General de las Clases.

Nombre: TReporte	
Tipo de clase: Controladora	
Atributo	Tipo
IdRep	Integer
Nombre	String
Estado	Integer
FI	TDateTime
FM	TDateTime
FF	TDateTime
IntSeg	Integer
Imp	Integer

Prog	Integer
TG	String
IG	String
IP	String
Para cada responsabilidad:	
Nombre:	Descripción
Crear()	Crear un Reporte tomando todos sus datos.
Abrir()	Abre un Reporte.
Modificar()	Modifica un Reporte.
Imprimir()	Imprime un Reporte Generado.
Exportar()	Exporta a Web un Reporte Generado.
Eliminar()	Elimina un Reporte.

Nombre: TVariables	
Tipo de clase: Controladora	
Atributo	Tipo
Nombre	String
IdRep	Integer
Idear	Integer
Valor	String
Prog	Integer
Para cada responsabilidad:	
Nombre:	Descripción
Crear()	Crear una Variable tomando sus datos.
Modificar()	Modifica una Variable.
Eliminar()	Elimina una Variable.

Nombre: TSistema	
Tipo de clase: Controladora	
Atributo	Tipo
Web	String
Clave	String
Tiempo	Integer
Para cada responsabilidad:	
Nombre:	Descripción
Modificar Datos()	Modifica los Datos del Generador.
Pedir Datos()	Pide los Datos del Generador.

Nombre: Thread_Actualizar_Reportes	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción

Actualizar Reportes()	Actualiza los Reportes en el Generador.

Nombre: Thread_Actualizar_Variables	
Tipo de clase: Controladora	
Para cada responsabilidad:	
Nombre:	Descripción
Actualizar Reportes()	Actualiza las Variables en el Generador.

2.7 Diseño de la Base de Datos.

Para diseñar la base de datos del sistema, se utilizó el diagrama de clases persistentes y el modelo de datos. Algunas de las clases representan los datos que se obtienen y almacenan durante los procesos de la aplicación, estos son los que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará el modelado de la lógica de negocio de la aplicación.

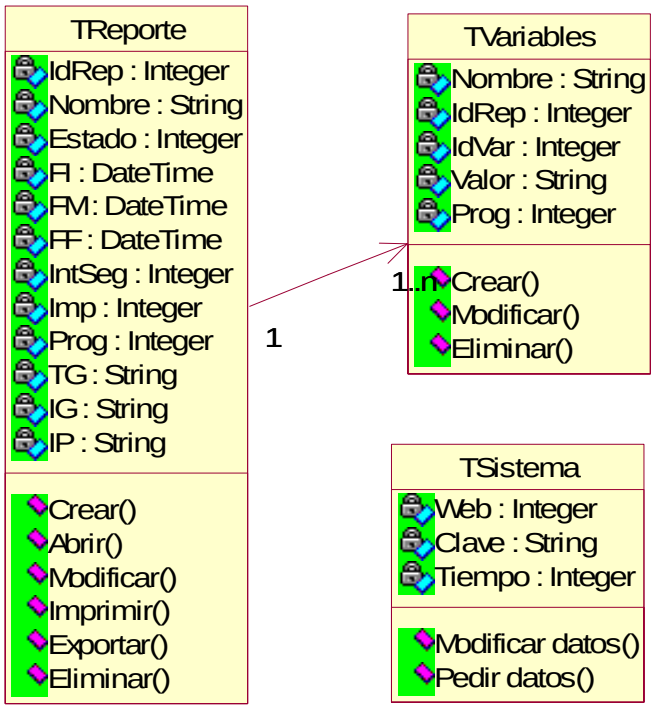

Figura 2.8 Diagrama de Clases Persistentes.

3.7.1 Modelo de Datos.

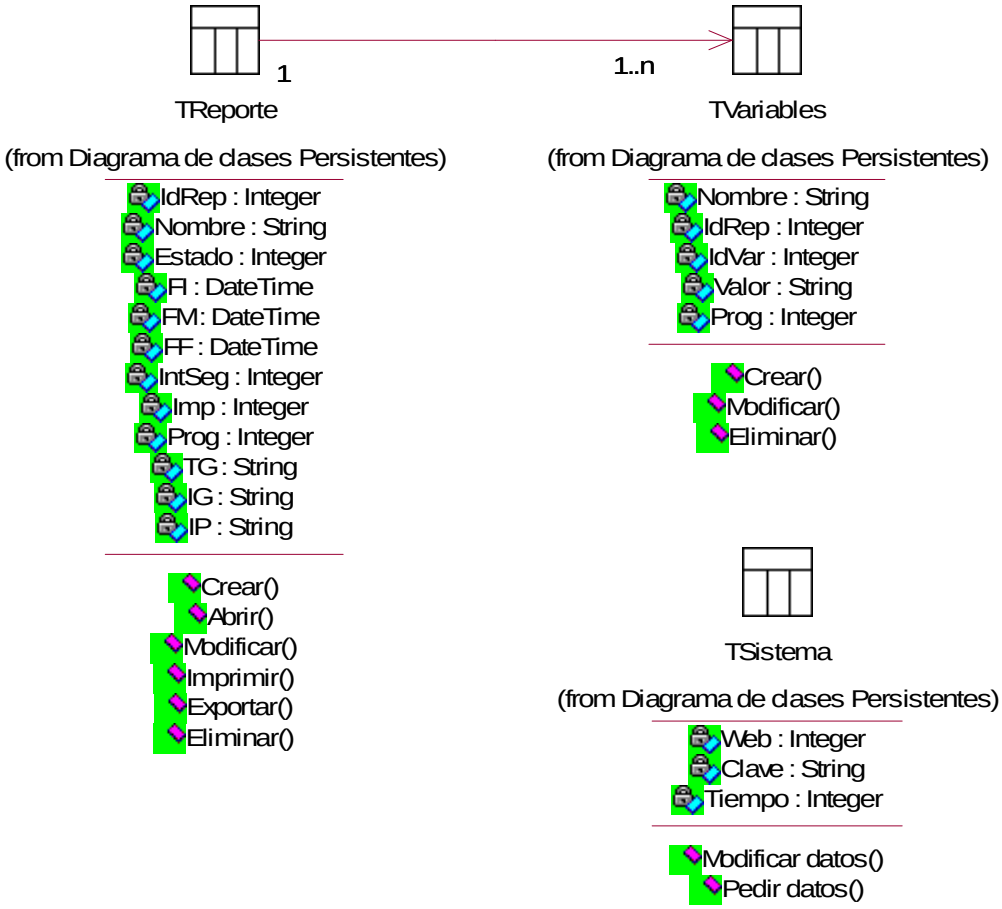

Figura 2.9 Modelo de Datos.

2.7.2 Descripción de las Tablas.

Nombre: TReporte		
Descripción: Es la tabla de los Reportes Procesados por el Generador.		
Atributo	Tipo	Descripción
IdRep	Integer	Es el identificador numérico del Reporte.
Nombre	String	Nombre del Reporte.
Estado	Integer	Estado del Reporte(Activo o No Activo).
FI	TDateTime	Fecha y Hora de Inicio del Reporte.
FM	TDateTime	Fecha y Hora de Modificación del Reporte.
FF	TDateTime	Fecha y Hora de Generación del Reporte.
IntSeg	Integer	Intervalo de Seguimiento dado en segundos.
Imp	Integer	Número de Copias a Imprimir.

Prog	Integer	Número de progresiones.
TG	String	Tipo de Generación.
IG	String	Intervalo de Generación.
IP	String	Intervalo de Progresión.

Nombre: TVariables		
Descripción: Es la tabla de las Variables correspondientes a los Reportes Procesados por el Generador.		
Atributo	Tipo	Descripción
IdRep	Integer	Es el identificador numérico del Reporte.
Nombre	String	Nombre de la Variable.
IdVar	Integer	Es el identificador numérico de la Variable.
Valor	String	Valor de la Variable.
Prog	Integer	Número de Progresión de la Variable.

Nombre: TSistema		
Descripción: Es la tabla de los Datos Específicos del Generador.		
Atributo	Tipo	Descripción
Web	Integer	Nos dice si se publicarán los reportes a Web.
Clave	String	Clave para el Modo Administrativo.
Tiempo	Integer	Tiempo en minutos del Modo Administrativo.

2.8 Diagrama de Despliegue.

La arquitectura física se compone de un nodo Generador de Reportes que aloja la aplicación interactuando con el nodo Servidor de Base de Datos donde se encuentran los datos de la aplicación. Para desarrollar toda esta funcionalidad, los nodos se interconectan a través de TCP/IP aunque el nodo Servidor de la Base de Datos puede ser el mismo nodo del Generador de Reportes.

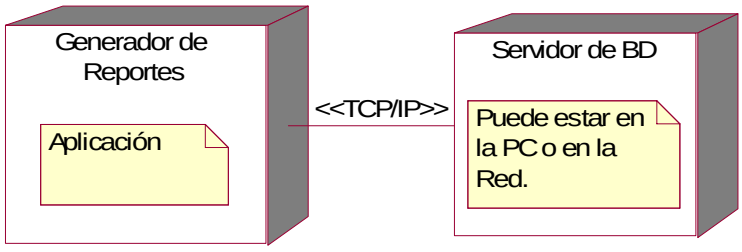

Figura 2.9 Diagrama de Despliegue.

2.9 Diagrama de Componentes.

Los diagramas de componentes se utilizan para modelar la vista de implementación del sistema. Esto implica modelar las cosas físicas que existen en un nodo, tales como ejecutables, bibliotecas, tablas, archivos y documentos. A continuación mostramos el diagrama de componentes de nuestro sistema.

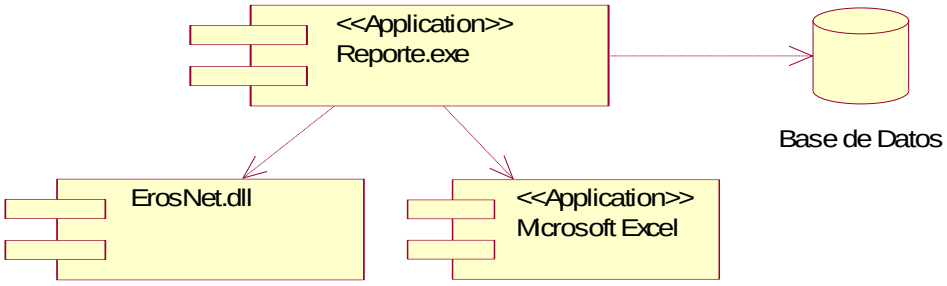

Figura 2.10 Diagrama de Componentes.

3

Capítulo

Estudio de Factibilidad.

Cualquier proyecto de software puede realizarse si el equipo que está a cargo de elaborarlo cuenta con recursos y tiempo infinito; pero lamentablemente no es así. La mayor parte de los proyectos informáticos presentan carencias de recursos y las fechas de entrega no se corresponden con la realidad. Es por ello que cada día se hace más necesaria la realización de estimaciones al inicio y a lo largo del ciclo de vida de los proyectos, aún cuando esta mirada al futuro tenga cierto grado de incertidumbre.

La estimación es la base de todas las demás actividades de planificación del proyecto y sirve como guía para una buena Ingeniería de Software. Aunque algunos autores plantean que es más un arte que una Ciencia, es una actividad importante que no debe restársele prioridad o relegarla a un último plano.

La pérdida de recursos, esfuerzo, tiempo y crédito profesional justifican la prudencia de evaluar la viabilidad de un proyecto cuanto antes y determinar si está mal concebido en la fase de definición.

En este capítulo se expone el estudio de factibilidad del proyecto, centrado en estimaciones de esfuerzo humano, tiempo de desarrollo para su ejecución y costo, realizadas con el método de puntos de función del modelo de COCOMO II en la etapa de diseño temprano. Se estiman los beneficios tangibles e intangibles que representan para el sistema propuesto, un análisis de costos y beneficios que permiten valorar si es factible el sistema.

3.1 Planificación.

La estimación del proyecto se realizó mediante los puntos de función desajustados, los cuales se utilizan para el cálculo de las instrucciones fuentes. De esta forma se estima la magnitud del

sistema y se obtienen además indicadores como: cantidad de hombres, el esfuerzo, el tiempo de duración y el costo del mismo.

3.2 Características del proyecto.

Se desglosan a continuación los requerimientos del sistema, los cuales se agrupan en: Entradas externas, Salidas externas, Peticiones, Ficheros internos, e Interfaces externas. Todos ellas se clasifican por su nivel de complejidad en: Simple, Media y Compleja.

3.3 COCOMO II.

Barry Boehm, en su libro clásico sobre economía de la Ingeniería del Software, introduce una jerarquía de modelos de estimación de Software con el nombre de COCOMO, por su nombre en Inglés (COnstructive, COst, MOdel) modelo constructivo de costos. Basados en estos modelos se estimarán el esfuerzo, tiempo de desarrollo, cantidad de hombres y costo de componentes representativos del software

3.3.1 Pasos para Calcular usando COCOMO II**.**

Pasos para la estimación.

1. Obtener los puntos de función. (UFP).

- Identificación de las características.
- Clasificación.
- Ponderación aplicando pesos.

2. Estimar la cantidad de instrucciones fuente. (SLOC).

- Utilizar tabla de lenguajes.

3. Aplicar las fórmulas de Bohem.

- Obtener esfuerzo (PM) y tiempo (TDEV).
- Costo del Proyecto

3.3.1.1. Obtención de los puntos de función. (UFP).

Identificar las Características.

- Entradas externas. EI.
- Salidas externas. EO.
- Ficheros lógicos internos. ILF.
- Ficheros de interfaz externa. ELF.
- Consultas (peticiones) externas. EQ.

No teniendo definido el Modelo de Análisis para el Caso de Estudio, se trabajó con los requerimientos funcionales del sistema, que es lo más cercano a los datos que se van a manipular.

Presentamos ahora los requerimientos:

1. Validar Modo. (Modo Administrativo o Modo Operativo).
2. Configurar Generador de Reportes.
3. Crear Modelo de Reporte.
4. Modificar Modelo de Reporte.
5. Eliminar Modelo de Reporte.
6. Ver Modelo de Reporte.
7. Crear Reporte.
8. Eliminar Reporte.
9. Ver Reporte.
10. Exportar Reporte Generado a Web.
11. Imprimir Reporte Generado.
12. Generar Reporte Configurado de forma Manual.

Cada requerimiento analizado en el paso anterior se clasificará y se le asignará un peso.

Tabla 3.1 Ficheros lógicos internos. ILF, Ficheros de interfaz externa. ELF.

ILF, ELF			
	Elementos de Datos		
Records	1 - 19	20 - 50	51+
1	Bajo	Bajo	Media
2 – 5	Bajo	Medio	Alto
6 +	Medio	Alto	Alto

Tabla 3.2 Salidas externas. EO. Consultas (peticiones) externas. EQ.

EO, EQ			
	Elementos de Datos		
Ficheros	1 - 5	6 - 19	20+
0,1	Bajo	Bajo	Media
2 – 3	Bajo	Medio	Alto
4 +	Medio	Alto	Alto

Tabla 3.3 Entradas externas. EI.

EI			
	Elementos de Datos		
Ficheros	1 - 4	5 - 15	16+
0,1	Bajo	Bajo	Media
2 – 3	Bajo	Medio	Alto
4 +	Medio	Alto	Alto

Al aplicar los pesos, se cuenta la cantidad de transacciones por cada Nivel de complejidad bajo y se multiplica por el peso asociado en la tabla 4. Todos estos productos se suman y se obtienen los puntos de función desajustados (UFP).

Tabla 3.4 Pesos según nivel de complejidad.

	Nivel de Complejidad		
Características	Bajo	Medio	Alto
ILF	7	10	15
ELF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

Entonces, para este caso obtenemos lo siguiente:

Tabla 3.5 Puntos de Función desajustados.

ELEMENTOS	SIMPLES		MEDIOS		COMPLEJOS		SUBTOTAL DE PUNTOS DE FUNCIÓN
	No	X Peso	No	X Peso	No	X Peso	
Ficheros lógicos internos	5	7					35
Entradas externas	3	3	2	4			17
Salidas externas	1	4	1	5			9
Peticiones	3	3	2	4			17
Total Puntos de Función Desajustados UFP.							**78**

3.3.1.2 Estimación de la cantidad de instrucciones fuente. (SLOC).

Para el cálculo de las instrucciones fuentes (SLOC) se utilizó la fórmula siguiente:

$$SLOC = UFP * ratio.$$

Luego:

SLOC = 78* 91

SLOC ≈ 7098 líneas de código fuente.

KSLOC= 7.098 (Miles de líneas de código)

Donde UFP es el total de puntos de función desajustados, y *ratio* es una constante para las SLOC de cada lenguaje de programación, en este caso tiene un valor para Borland Delphi de 91.

3.3.1.3 Aplicación de las Fórmulas de Bohem.

Luego de calculada la cantidad de instrucciones fuentes, se utilizó este valor en el cálculo del esfuerzo dado por la fórmula de Bohem:

Obtener esfuerzo (PM) y tiempo de desarrollo (TDEV).

$$PM_{NS} = A \times Size^{E} \times \prod_{i=1}^{n} EM_i \qquad TDEV_{NS} = C \times (PM_{NS})^{F}$$

$$\text{donde} \quad E = B + 0.01 \times \sum_{j=1}^{5} SF_j \qquad \text{donde} \quad F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SF_j$$

$$= D + 0.2 \times (E - B)$$

Figura 3.1 Fórmula de Bohen.

Donde:

Size: Tamaño estimado (KSLOC).

A = 2.94, B = 0.91, C = 3.67, D = 0.28

Se tiene el tamaño (KSLOC) faltaría calcular E el cual depende de los factores de escala:

Son cinco factores que afectan E, (el exponente del TAMAÑO).

De Acuerdo con los Factores de escala y utilizando los enumerados que se presentan a continuación se puede escoger el valor para cada factor.

SFj: Factores de Escala (10)

PREC: Precedencia.

FLEX: Flexibilidad.

RESL: Riesgos.

TEAM: Cohesión del Equipo.

PMAT: Madurez de las Capacidades.

PREC: Desarrollos previos similares

0.00, es idéntico a previos	1.24, es muy parecido	2.48, bastante parecido
3.72, aspectos novedosos	4.96, muy diferente	6.2, totalmente diferente.

FLEX: Flexibilidad del desarrollo (e.g. grado de acuerdo con requerimientos preestablecidos o con interfaces externos preexistente)

0.0, metas son generales	1.01, cierto acuerdo	2.03, acuerdo general
3.04, cierta flexibilidad	4.05, flexibilidad ocasional	5.07, riguroso

RESL: Manejo de riesgos y arquitectura.

Tabla 3.6 Valor y Justificación de Manejo de riesgos y arquitectura.

VALOR	JUSTIFICACIÓN
0.0	Plan, identifica todos los riesgos críticos y establece hitos para resolverlos, calendario y presupuesto toma en cuenta riesgos, arquitectura puede tomarse hasta el 40% del esfuerzo de desarrollo, herramientas disponibles para resolver/mitigar riesgos y verificar especificidad de la arquitectura, muy poca incertidumbre, interfaz con usuario, tecnología, desempeño, riesgos no son críticos;
1.41	Plan, identifica la mayoría de los riesgos críticos y establece hitos para resolverlos, calendario y presupuesto toma en cuenta la mayoría de los riesgos, arquitectura puede tomarse hasta el 33% del esfuerzo de desarrollo, herramientas disponibles para resolver/mitigar mayoría de riesgos y verificar especificidad de la arquitectura, poca incertidumbre, interfaz con usuario, tecnología, desempeño, riesgos no son críticos.
2.83	Plan, identifica muchos de los riesgos críticos y establece hitos para resolverlos, calendario y presupuesto generalmente toma en cuenta riesgos, arquitectura puede tomarse hasta el 25% del esfuerzo de desarrollo, herramientas regularmente disponibles para resolver/mitigar riesgos y verificar especificidad de la arquitectura, algo de incertidumbre, interfaz con usuario, tecnología, desempeño, no más de un riesgo crítico.
4.24	Plan, identifica algunos de los riesgos críticos y establece hitos para resolverlos, calendario y presupuesto toma en cuenta algunos de los riesgos, arquitectura puede tomarse hasta el 17% del esfuerzo de desarrollo, hay problemas con la disponibilidad del arquitecto, algo de herramientas disponibles para resolver/mitigar riesgos y verificar especificidad de la arquitectura, considerable incertidumbre, interfaz con usuario, tecnología, desempeño, entre 2-4 riesgos críticos.
5.65	Plan, identifica pocos riesgos críticos y establece hitos para resolverlos, calendario y presupuesto toma en cuenta pocos riesgos, arquitectura puede tomarse hasta el 10% del esfuerzo de desarrollo, hay problemas con la disponibilidad del arquitecto (disp. menor al 40%), pocas herramientas disponibles para resolver/mitigar riesgos y verificar especificidad de la

	arquitectura, significativa incertidumbre, interfaz con usuario, tecnología, desempeño, entre 5-10 riesgos críticos
7.07	Plan, no identifica los riesgos críticos, calendario y presupuesto no toma en cuenta los riesgos, arquitectura puede tomarse hasta el 5% del esfuerzo de desarrollo, hay problemas con la disponibilidad del arquitecto (disp. menor del 20%), herramientas no disponibles para resolver/mitigar riesgos y verificar especificidad de la arquitectura, extrema incertidumbre, interfaz con usuario, tecnología, desempeño, más de 10 riesgos críticos

TEAM: Cohesión del equipo de desarrollo.

Tabla 3.7 Valor y Justificación de Cohesión del equipo de desarrollo.

VALOR	JUSTIFICACIÓN
0.0	Interacciones fluidas, objetivos y culturas de accionistas totalmente consistentes, total habilidad y disponibilidad de accionistas para acomodar objetivos de otros accionistas, dilatada experiencia previa operando como equipo, visión y compromisos 100% compartidos.
1.1	Interacciones altamente cooperativas, objetivos y culturas de accionistas fuertemente consistentes, fuerte habilidad y disponibilidad de accionistas para acomodar objetivos de otros accionistas, considerable experiencia previa operando como equipo, visión y compromisos considerablemente compartidos.
2.19	Interacciones principalmente cooperativas, objetivos y culturas de accionistas considerablemente consistentes, considerable habilidad y disponibilidad de accionistas para acomodar objetivos de otros accionistas, mediana experiencia previa operando
3.29	Interacciones básicas cooperativas, objetivos y culturas de accionistas básicamente consistentes, habilidad y disponibilidad básica de accionistas para acomodar objetivos de otros accionistas, poca experiencia previa operando como equipo, visión y compromisos poco compartidos.
4.38	Algunas interacciones difíciles, objetivos y culturas de accionistas algo consistentes, algo habilidad y disponibilidad de accionistas para acomodar objetivos de otros accionistas, poca experiencia previa operando como equipo, visión y compromisos poco compartidos.
5.48	Interacciones difíciles, objetivos y culturas de accionistas poco consistentes, poca habilidad y disponibilidad de accionistas para acomodar objetivos de otros accionistas, nada de experiencia previa operando como equipo, visión y compromisos nada compartidos.

PMAT: nivel de madurez estimada, en relación al modelo de madurez de software CMM:

0.00, nivel 5	1.56, nivel 4	3.12, nivel 3
4.68, nivel 2	6.24, nivel 1, superior	7.80, nivel 1, inferior.

3.3.1.4 Cálculo del Esfuerzo.

El esfuerzo es la cantidad de tiempo que una persona invierte trabajando en el desarrollo de un proyecto durante un mes. La sigla que lo representa es PM.

Tabla 3.8 Constantes y fórmulas para el cálculo del esfuerzo.

SIGLAS	INDICADOR	VALOR O FÓRMULA
PM	Esfuerzo	$A * (Size)^{E} * \prod EMi$
A	Constante	2.94
Size	Miles de instrucciones fuentes	7.098
E	Agregado de 5 factores de escala	$B + 0.01 * \Sigma SFi$
EM	Multiplicadores de esfuerzo	Se muestran en la tabla 3.10
B	Constante	0.91
SF	Factores de escala	Se muestran en la tabla 3.9

Relacionados con el producto:

RELY: fiabilidad exigida al software,

DATA: tamaño de la base de datos (aunque parece más relacionado con el esfuerzo necesario para capturar los datos de pruebas)

CPLX: complejidad del producto;

RUSE: desarrollo para ser reutilizado;

DOCU: cantidad de artefactos que deben ser documentados;

Relacionados con la plataforma de desarrollo:

TIME: Exigencias sobre capacidad de ejecución;

STOR: Exigencias sobre almacenamiento del sistema;

PVOL: Volatilidad de la plataforma, se considera muy volátil una plataforma que cambia cada dos semanas, poco volátil una que cambie cada doce meses);

Relacionados con personal:

ACAP: capacidad de los analistas;

PCAP: capacidad de los programadores;

PCON: volatilidad del personal;

APEX: experiencia previa en área de aplicación;

PLEX: experiencia previa con la plataforma;

LTEX: experiencia previa con el lenguaje y herramientas de desarrollo;

Relacionados con el proyecto:

TOOL: uso de herramientas de software;

SITE: desarrollo en localidades distribuidas;

SCED: exigencias sobre el calendario.

Ajustes al modelo base por estimación más temprana (diseño pre-arquitectura)

COCOMO II sugiere que muchos de los 17 multiplicadores de esfuerzo no pueden estimarse en el diseño temprano, por lo que los reduce a 7 multiplicadores:

RCPX: Confiabilidad y complejidad del producto.

RUSE: Nivel de reutilizabilidad del desarrollo.

PDIF: Dificultad de uso de la plataforma.

PERS: Capacidad del personal de desarrollo.

PREX: Experiencia del personal de desarrollo.

FCIL: Facilidades de desarrollo.

SCED: exigencias sobre el calendario.

Tabla 3.9 Factores de Escala.

FACTOR DE ESCALA	VALOR
PREC	3.72
FLEX	3.04
RESL	4.24
TEAM	3.29
PMAT	4.68

Tabla 3.10 Multiplicadores de Esfuerzo.

MULTIPLICADOR (+)	VALOR
RCPX	1.00
RUSE	1.15
PDIF	1.00
PERS	0.63
PREX	1.22
FCIL	0.73
SCED	1.00

Para obtener los resultados de las fórmulas anteriormente expuestas, se calcularon los valores de cada factor de escala (SF_j) y de cada multiplicador de esfuerzo (EM_i).

Producto de los multiplicadores de esfuerzo:

ΠEM = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED

ΠEM= 1.00 * 1.15 * 1.00 * 0.63 * 1.00 * 0.73 * 1.00

ΠEM= 0,528885

Sumatoria de los factores de escala:

ΣSF = PREC + FLEX + RESL + TEAM + PMAT

ΣSF = 3.72 + 3.04 + 4.24 + 3,29 + 4,68

ΣSF= 18,97

Cálculo de esfuerzo

E= B + 0.01 * Σ SFi

E= 0.91 + 0.01 * 18.97 = 1,0997

$PM = A * (Size)^{E} * \prod EMi$

$PM= 2.94 * 7.098^{1,0997} * 0,528885$= 13,41 hombres-mes.

Se necesitan 14 personas para realizar el software en un mes.

3.3.1.5 Determinación del Tiempo de Desarrollo.

Al saberse el valor del esfuerzo se puede calcular el tiempo de desarrollo (TDEV) estimado del software, es decir, cantidad de meses necesarios para desarrollar el software.

Tabla 3.11 Constantes y fórmulas para el cálculo del tiempo de desarrollo.

SIGLAS	INDICADOR	VALOR O FÓRMULA
TDES	Tiempo de desarrollo.	$C * (PM)^{F}$
C	Constante.	3.67
PM	Esfuerzo.	13,41 hombre-mes
F	Exponente de escala.	D + 0.2 * (E - B)
D	Exponente base para la ecuación del cronograma (constante).	0.28
E	Agregado de 5 factores de escala	B + 0.01 * Σ SFi
B	Exponente de base escalado para la ecuación de esfuerzo que puede ser calibrado (constante).	0.91
ΣSF	Factores de escala	18,97

$F= D + 0.2 * (E - B)$

$F = 0.28 + 0.2 (1,0997- 0.91) = 0,31794$

$TDES= C * (PM)^{F}$

$TDES = 3.67 * (13,41)^{0.31794} = 8,37$ meses

EL tiempo necesario para desarrollar el proyecto es de 8,37 meses.

3.3.1.6 Determinación de la Cantidad de Hombres.

Teniendo en cuenta el tiempo de desarrollo se calcula la cantidad de hombres (CH) necesarios para desarrollar el software, se obtiene la Tabla siguiente:

Tabla 3.12 Constantes y fórmulas para el cálculo de la cantidad de hombres.

SIGLAS	INDICADOR	VALOR O FÓRMULA
CH	Cantidad de hombres por mes	PM/TDES
PM	Esfuerzo	13.41 hombre-mes
TDES	Tiempo de desarrollo	8.37 meses

CH = 13.41/ 8.37 = 1.6 hombres.

Son necesarios 2 hombres para realizar el software en 8.37 meses. Como en realidad trabaja una sola persona se reajustan los cálculos para este valor:

CH^{*}= 1 persona

TEDV = PM/ CH^{*} = 13.41/ 1 = 13.41 meses

Es necesarios 14 meses para que 1 persona desarrolle el software.

3.3.1.7 Determinación del Costo del Software.

El costo del software depende del salario promedio de las personas que lo desarrollan y del esfuerzo que ellas realizan para la ejecución del mismo y se calcula a través de la fórmula representada en la tabla siguiente:

Tabla 3.13 Constantes y fórmulas para el cálculo del costo de software.

SIGLAS	INDICADOR	VALOR O FÓRMULA
C	Costo del proyecto	CHM *PM
CHM	Costo de hombres por mes	CH^{*} * SP
SP	Salario básico de un Ingeniero	$ 225.00
PM	Esfuerzo	13.41 hombre-mes

Costo del software:

El salario medio es de $ 225.00

C = 1 *225 * 13,41 = $ 3017.25

El software cuesta $ 3017.25.

Tabla 3.13 Resultados de las estimaciones de esfuerzo, tiempo de desarrollo, cantidad de hombres y costo del proyecto.

CÁLCULO DE:	VALOR	JUSTIFICACIÓN
Esfuerzo	13.41 hombres-mes	Cantidad de tiempo que una persona invierte trabajando en el desarrollo de un proyecto
Tiempo de desarrollo	14 meses	Cantidad de meses para terminar el proyecto.
Cantidad de personas	1	Cantidad de personas necesarias para terminar el proyecto en 14 meses.
Costo	$ 3017.25	Cantidad de dinero que cuesta el proyecto después de terminado.
Salario medio	$ 225.00	Salario básico de un ingeniero.

3.4 Beneficios Tangibles e Intangibles.

Mediante el software es posible un óptimo aprovechamiento del tiempo en la generación de reportes y la toma de decisiones en las Industrias automatizadas con el Sistema de Supervisión y Control de Procesos EROS, dándole un mejor uso a las tecnologías de la información y las

comunicaciones (TIC). El proceso de creación de reportes por los usuarios lo pueden controlar y gestionar con mejores resultados. Se facilitan de forma considerable los procesos relacionados con la adquisición de reportes y uso de los datos, debido a que se organizan y manejan desde un sistema cliente de las estaciones de medición que cumple con los requisitos requeridos de seguridad y confiabilidad.

3.5 Análisis de Costos y Beneficios.

El proceso de desarrollo del sistema no supone gastos de recursos, porque se cuenta con la tecnología a utilizar. En cuanto a la implantación, sólo se necesita de una computadora para el funcionamiento óptimo y atendiendo a la factibilidad es importante señalar que un balance entre los costos y beneficios se inclina indiscutiblemente a la aplicación de la propuesta como solución.

Conclusiones

De acuerdo con los objetivos propuestos, se puede afirmar que el presente trabajo de diploma intentó más que dar una solución a la problemática de construir un Generador de Reportes que cumpliera con la necesidad de los usuarios actuales del Sistema de Supervisión y Control de Procesos EROS, abrir las puertas con el análisis y el estudio realizado a que en un futuro inmediato se pueda realizar la comprobación del alcance de los resultados.

El Generador de Reportes implementado permite como resultado que los operadores, ingenieros, supervisores y directivos puedan recibir sus reportes de la red industrial, aumentando el nivel de información del proceso y grado de eficiencia en la toma de decisiones. Fue concebido además, para que su código pueda ser modificado y adaptado a otra fuente para la adquisición de datos que no sea la del EROS.

Recomendaciones

Con vista al desarrollo futuro del Generador de Reportes para el Sistema de Supervisión y Control de Procesos EROS se recomienda:

- Realizar el análisis del alcance de los resultados obtenidos.
- Permitir que los reportes generados, además de tener la opción de ser impresos o publicados en Web, se puedan enviar por correo electrónico.
- Desarrollar el Generador de Reportes para que se puedan crear reportes que se generen a partir de un determinado comportamiento de una variable (Reportes por Eventos).
- Utilizar el Generador en todas las Industrias Automatizadas con el Sistema de Supervisión y Control de Procesos EROS.

Bibliografía

[1]. Advantech. http://www.advantech.com (19/03/06).

[2]. eMation. http://www.emation.com (21/03/06).

[3]. Iconics. http://www.iconics.com (21/03/06).

[4]. National Instruments. http://www.ni.com (21/03/06).

[5]. Rockwell Automation. http://www.software.rockwell.com (10/04/06).

[6]. TA-Engineering Product. http://www.ta-eng.com/home.htm (10/04/06).

[7]. Ivar Jacobson,Grady Booch, James Rumbaugh. “El Proceso Unificado de Desarrollo del Software”, La Habana, 2004, Editorial Felix Varela ,Volumen I,II.

[8]. Booch, G., Rumbaugh, J., Jacobson, I. El Lenguaje Unificado de Modelado. Addison-Wesley. 1999.

[9]. Craig Larman, “UML y Patrones”,La Habana, 2004, Editorial Felix Varela, Volumen I,II.

[10]. Diagramas de Casos de Uso. http://www.vico.org/MuestrarioDiagCU.pdf (20/05/06).

Glosario de Términos.

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con el sistema o negocio.

Proceso: Secuencia de actividades invocadas para producir un producto de software.

Protocolo: Descripción formal de formatos de mensaje y de reglas que dos ordenadores deben seguir para intercambiar dichos mensajes. Un protocolo puede describir detalles de bajo nivel de las interfaces máquina-a-máquina o intercambios de alto nivel entre programas de asignación de recursos.

Rol: Papel, cometido o función que tiene o desempeña un actor.

RUP: El Proceso Unificado Rational (RUP) es una metodología de desarrollo para la programación orientada a objetos. Según Rational (diseñadores de Rose Rational y el Idioma Modelado Unificado), RUP está como un mentor en línea que mantiene pautas, plantillas, y ejemplos de todos los aspectos y fases de desarrollo del programa. RUP es un software comprensivo que diseña herramientas que combinan los aspectos procesales de desarrollo (como las fases definidas, técnicas, y prácticas) con otros componentes de desarrollo (como los documentos, modelos, manuales, el código, y así sucesivamente) dentro de una armazón unificándose.

Software:*(Componentes lógicos, programas, software)*. Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.

TCP/IP: (*Transmission Control Protocol/Internet Protocol)* Es el conjunto de protocolos que definen a Internet. Originalmente diseñado para el sistema operativo UNIX, hoy día existe software TCP/IP disponible para la mayoría de los sistemas operativos. Para poder utilizar la Internet, su computador debe tener software TCP/IP.

UML: Lenguaje unificado de modelado -Unified Modeling Language.

Usuario: Persona que usa ordinariamente una cosa.

Anexos.

Anexo 1. Descripción Textual de lo Casos de Usos del Sistema.

Nombre del caso de uso	Validar Modo.
Actores	Operador (inicia).
Propósito	Validar el acceso al Modo Administrativo.
Resumen. El Operador coloca la contraseña para Abrir el Modo Administrativo.	
Referencias	R1
Precondiciones	El Generador debe estar en Modo Operativo.
Poscondiciones	Queda Abierto el Modo Administrativo.

Nombre del caso de uso	Configurar Generador.
Actores	Administrador (inicia).
Propósito	Configurar la Administración del Generador de Reportes.
Resumen. El Administrador debe definir la contraseña y el tiempo de duración para el Modo Administrativo y si se publicarán o no automáticamente los reportes en formato Web.	
Referencias	R2 Validar Modo<<include>>
Precondiciones	El Generador debe estar en Modo Administrativo.
Poscondiciones	Queda configurada la Administración del Generador de Reportes.

Nombre del caso de uso	Crear Modelo de Reporte.
Actores	Administrador (inicia).
Propósito	Crear un modelo de reporte.
Resumen. El Administrador debe definir como será el modelo de reporte aunque después pueda modificarlo.	
Referencias	R3 Validar Modo<<include>>
Precondiciones	El Generador debe estar en Modo Administrativo.
Poscondiciones	Queda configurado el modelo de reporte.

Nombre del caso de uso	Modificar Modelo.
Actores	Administrador (inicia).
Propósito	Modificar un modelo de reporte.
Resumen. El Administrador debe modificar como será el modelo de reporte acorde a sus necesidades.	

Referencias	R4 Validar Modo<<include>>
Precondiciones	El Generador debe estar en Modo Administrativo.
Poscondiciones	Queda modificado el modelo de reporte.

Nombre del caso de uso	Eliminar Modelo.
Actores	Administrador (inicia).
Propósito	Eliminar un modelo de reporte.
Resumen. El Administrador debe seleccionar el modelo que desea eliminar y este es eliminado.	
Referencias	R5 Validar Modo<<include>>
Precondiciones	El Generador debe estar en Modo Administrativo y el modelo no puede estar en uso.
Poscondiciones	Queda eliminado el modelo de reporte.

Nombre del caso de uso	Ver Modelo.
Actores	Operador o Administrador (inicia).
Propósito	Ver un Modelo de Reporte del Generador.
Resumen. El Operador o el Administrador del Generador seleccionan el modelo de reporte que desean ver y el generador se los muestra.	
Referencias	R6
Precondiciones	
Poscondiciones	Queda mostrado el modelo de reporte seleccionado.

Nombre del caso de uso	Crear Reporte.
Actores	Administrador (inicia).
Propósito	Crear un reporte.
Resumen. El Administrador selecciona el modelo configurado con anterioridad para el nuevo reporte y de este modo es creado.	
Referencias	R7 Validar Modo<<include>>
Precondiciones	El Generador debe estar en Modo Administrativo.
Poscondiciones	Queda creado el reporte.

Nombre del caso de uso	Eliminar Reporte.
Actores	Administrador (inicia).
Propósito	Eliminar un reporte.
Resumen. El Administrador selecciona el reporte que desea eliminar y este es eliminado.	
Referencias	R8

	Validar Modo<<include>>
Precondiciones	El Generador debe estar en Modo Administrativo.
Poscondiciones	Queda eliminado el reporte.

Nombre del caso de uso	Ver Reporte.
Actores	Operador o Administrador (inicia).
Propósito	Ver un Reporte del Generador.
Resumen. El Operador o el Administrador del Generador seleccionan el reporte que desean ver y el generador se los muestra.	
Referencias	R9
Precondiciones	
Poscondiciones	Queda mostrado el reporte seleccionado.

Nombre del caso de uso	Exportar Reporte Web.
Actores	Operador o Administrador (inicia).
Propósito	Exportar un reporte generado a Web.
Resumen. El Operador o el Administrador del Generador seleccionan el reporte generado que desean exportar a Web y el reporte es exportado.	
Referencias	R10
Precondiciones	
Poscondiciones	Queda exportado a Web el reporte generado seleccionado.

Nombre del caso de uso	Imprimir Reporte.
Actores	Operador o Administrador (inicia).
Propósito	Imprimir un reporte generado.
Resumen. El Operador o el Administrador del Generador seleccionan el reporte generado que desean imprimir y el mismo es impreso por la impresora predeterminada.	
Referencias	R11
Precondiciones	
Poscondiciones	Queda impreso el reporte generado seleccionado.

Nombre del caso de uso	Generar Reporte Manual.
Actores	Operador o Administrador (inicia).
Propósito	Generar de forma manual un reporte configurado.
Resumen. El Operador o el Administrador del Generador seleccionan el reporte configurado que desean generar manualmente y el mismo es generado.	
Referencias	R12
Precondiciones	
Poscondiciones	Queda generado el reporte configurado seleccionado.

Anexo 2. Diagramas de Secuencia de los Casos de Usos el Sistema.

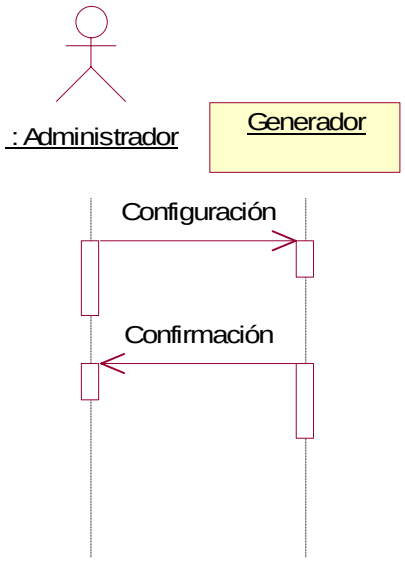

Diagrama de Secuencia del caso de uso Configurar Generador.

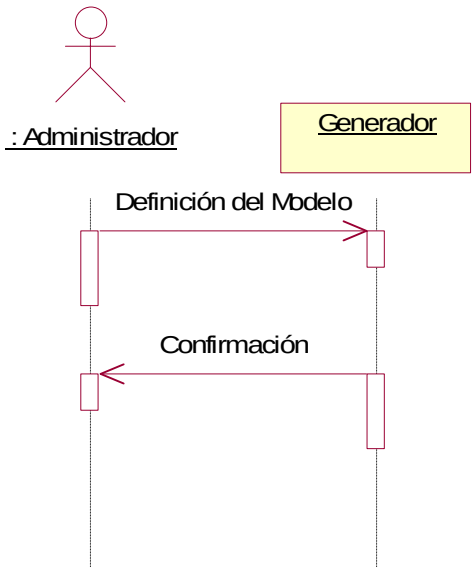

Diagrama de Secuencia del caso de uso crear Modelo.

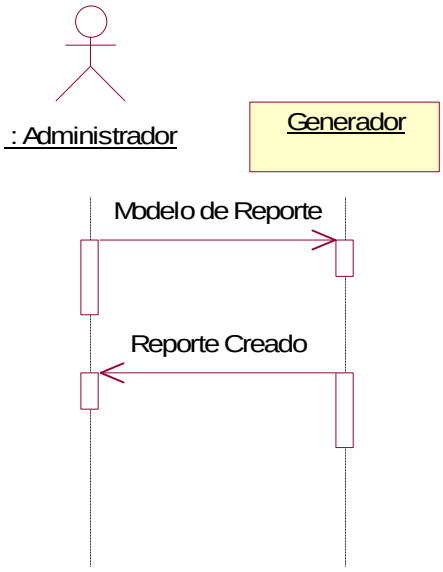

Diagrama de Secuencia del caso de uso Crear Reporte.

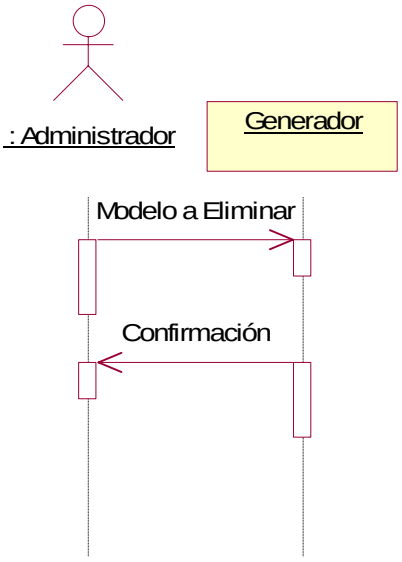

Diagrama de Secuencia del caso de uso Eliminar Modelo.

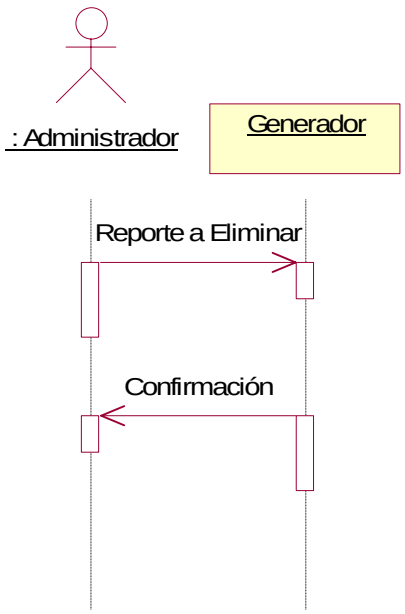

Diagrama de Secuencia del caso de uso Eliminar Reporte

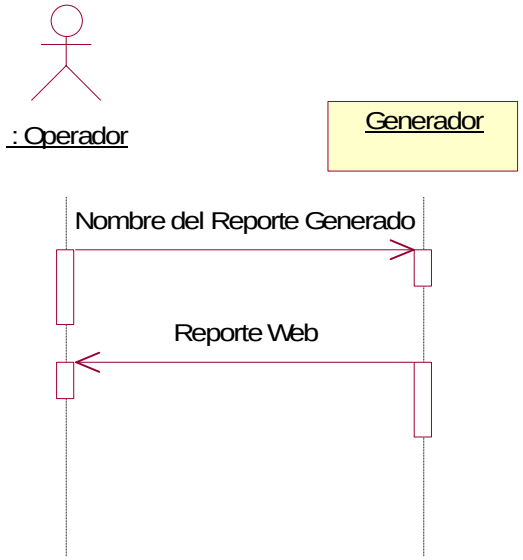

Diagrama de Secuencia del caso de uso Exportar Reporte Web.

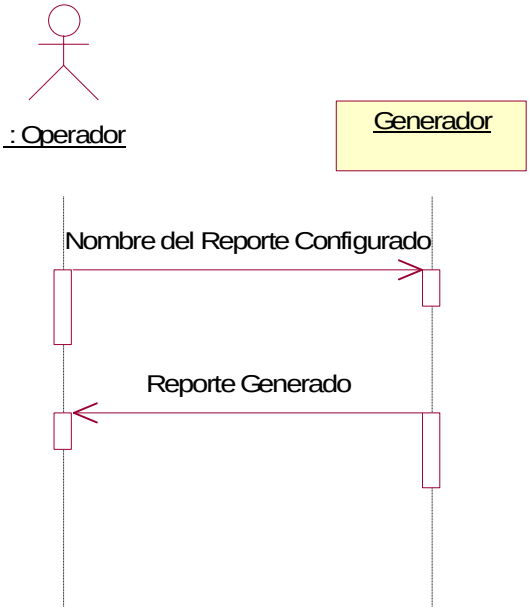

Diagrama de Secuencia del caso de uso Generar Reporte Manual.

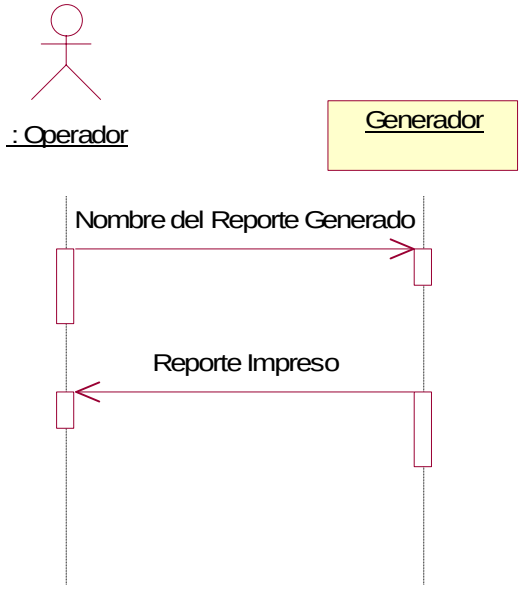

Diagrama de Secuencia del caso de uso Imprimir Reporte.

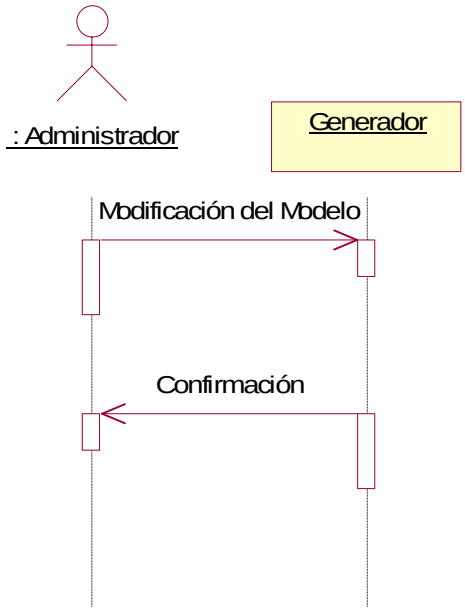

Diagrama de Secuencia del caso de uso Modificar Modelo.

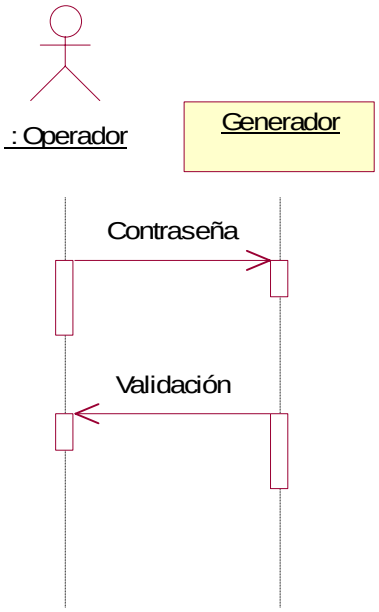

Diagrama de Secuencia del caso de uso Validar Modo.

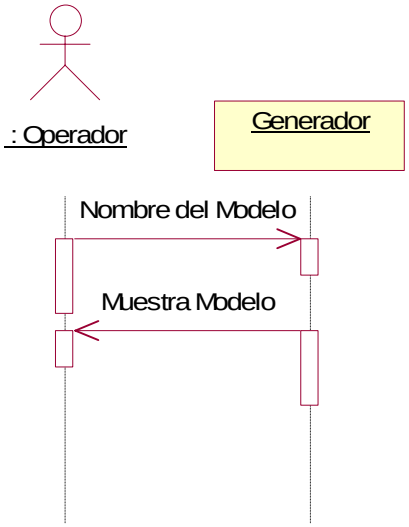

Diagrama de Secuencia del caso de uso Ver Modelo.

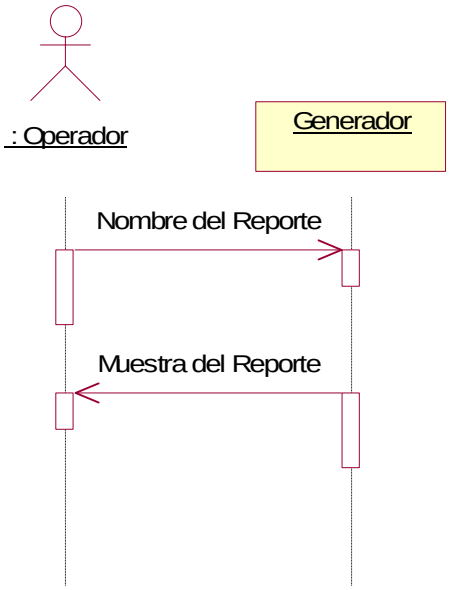

Diagrama de Secuencia del caso de uso Ver Reporte.