

**Universidad de las Ciencias Informáticas**

**Facultad 5**



**CreativeReport:** herramienta para reportes del sistema virtual de información de la Escuela Internacional de Cine y Televisión (EICTV).

**Trabajo de Diploma para optar por el título de  
Ingeniera en Ciencias Informáticas**

**Autor(a):** Danelys Inerarity Alegre

**Tutora:** Lic. Estrella Hendrickson

**Co-tutor:** Ing. Mario Mayor.

**Cuidad de La Habana, julio de 2010**



## **Declaración de Autoría**

Declaro que soy la única autora de este trabajo y autorizo a la Escuela Internacional de Cine y Televisión (EICTV) y al área de la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Autor(a)

---

Tutor(es)

## Datos de Contacto

**Autor(a):** Danelys Inerarity Alegre [dinerarity@estudiantes.uci.cu](mailto:dinerarity@estudiantes.uci.cu)

**Tutor(a):** Lic. Estrella Hendrickson.

J´Dpto de informática de la EICTV.

**Co-tutor(a):** Ing. Mario Mayor, informático de la EICTV.

## **Agradecimientos**

Agradezco a mis padres todo el apoyo que me han dado, la convicción de que se podía lograr a pesar de todos los contratiempos. A ellos muchas gracias, de corazón, realmente no tengo palabras para retribuirles todo lo que han hecho por mí.

Agradecimientos a Estrella que sin su ayuda no hubiese sido posible este hecho, a mi pareja Héctor por su comprensión y dedicación y a todas aquellas personas que de una forma u otra hicieron posible realizar mi sueño.

## **Dedicatoria**

A mis padres Lucía y Alberto,  
A mi abuela Yoya, donde quiera que estés.  
A mi pareja Héctor,  
A todos ustedes este triunfo  
Y el corazón...

## **Resumen**

En la actualidad la EICTV de San Antonio de Los Baños cuenta con una tecnología avanzada en equipos del cine y de la informática, y de esta última con sistemas altamente desarrollados para la comodidad de trabajadores y estudiantes; por lo que en el departamento de informatización de la entidad, surge la idea de crear un sitio web, con interfaz amigable y fácil de interactuar, integrando de manera general toda la información de la escuela. En aras de esta situación se necesita de un software que su principal objetivo se convierta en generar reportes y mostrar en la web su contenido.

Se mencionan características relevantes y comunes de los generadores de reportes de la actualidad, donde se escogen algunas de éstas para desarrollar CreativeReport, de forma tal que responda a las necesidades de la EICTV.

### **Palabras Claves**

Tecnología, informática, sistema, intranet, información, herramienta, reportes, EICTV.

# Tabla de Contenidos

<b>Agradecimientos.....</b>	<b>5</b>
<b>Resumen.....</b>	<b>7</b>
<b>Introducción.....</b>	<b>10</b>
<b>Capítulo1: Fundamentación Teórica.....</b>	<b>15</b>
1.1 Rasgos generales del funcionamiento de generadores de reportes.....	15
1.2 Diseño de informes.....	16
1.3 Características de varios generadores de reportes.....	16
1.3.1 JasperReports.....	16
1.3.2 NCRReport.....	17
1.3.3 Ruport.....	18
1.3.4 CrystalReports.....	19
1.3.5 PHPReports.....	19
1.4 Principales Tendencias.....	19
1.5 Conclusiones.....	21
<b>Capítulo 2: Construcción de la solución propuesta.....</b>	<b>22</b>
2.1 El sistema virtual de la EICTV en la actualidad.....	22
2.1.1 Modelo de negocio.....	22
2.1.2 Actores del negocio.....	23
2.1.3 Trabajadores del negocio.....	23
2.1.4 Reglas del negocio.....	24
2.1.5 Modelo de casos de uso del negocio.....	24
2.1.6 Descripción textual del caso de uso del negocio Realizar informe.....	26
2.1.7 Diagrama de actividades.....	26



2.1.8 Requisitos funcionales.....	28
2.1.9 Requisitos no funcionales.....	28
2.2. Propuesta.....	29
2.3 Métodos de validación para XML.....	30
2.4 Modelo del sistema.....	35
2.4.1 Actores del sistema.....	35
2.4.2 Diagrama de casos de uso d el sistema.....	36
2.5 Modelo de análisis.....	36
2.5.1 Modelo de clases del análisis.....	37
2.5.2 Descripción de clases del análisis.....	37
2.5.3 Diagrama de análisis del sistema.....	39
2.6 Modelo de diseño.....	41
2.7 Diagrama de despliegue.....	42
2.8 Vistas del reporte creado.....	43
2.9 Conclusiones.....	47
<b>Capítulo 3: Validación de CreativeReport.....</b>	<b>48</b>
3.1 Casos de prueba. Características.....	48
3.1.1 Rasgos generales del caso de prueba caja negra.....	49
3.2 Pruebas de unidad realizadas a las principales funcionalidades del software.....	50
3.2.1 Caso de prueba del método <u>renderPreview ()</u> .....	50
3.2.2 Caso de prueba del método: <u>render ()</u> .....	51
3.3 Conclusiones.....	52
<b>Conclusiones.....</b>	<b>53</b>
<b>Recomendaciones.....</b>	<b>54</b>
<b>Referencias bibliográficas.....</b>	<b>55</b>
<b>Bibliografía.....</b>	<b>56</b>
<b>Glosario de términos.....</b>	<b>57</b>

# Introducción

A lo largo de la historia, la forma de almacenamiento y acceso a la información ha ido variando, en la edad media, el principal acervo se encontraba en las bibliotecas de los monasterios. A partir de la edad moderna, gracias al nacimiento de la imprenta, los libros comenzaron a fabricarse en serie y surgieron los periódicos. Este proceso de desarrollo ascendente no sólo ocurre en el ámbito social sino también en el mundo de la informática, que durante el siglo pasado, el avance devino en que actualmente se ligara la información a una verdadera revolución, especialmente concebida por la globalización e Internet.

La evolución de Internet como red de comunicación global y el surgimiento y desarrollo de la web como servicio imprescindible para compartir información, creó un excelente espacio para la interacción del hombre con la información hipertextual. Los sitios web, como expresión de sistemas de información, deben poseer los siguientes componentes:

- Usuarios.
- Mecanismos de entrada y salida de la información.
- Almacenes de datos, información y conocimiento.
- Mecanismos de recuperación de información.

Pudiésemos definir entonces como **sistema de información** al conjunto organizado de elementos, que pueden ser personas, datos, actividades o recursos materiales en general. Estos elementos interactúan entre sí para procesar información y distribuirla de manera adecuada en función de los objetivos de una organización.

Actualmente, estos sistemas se encuentran al alcance de las grandes masas de usuarios; en el que interactúan directamente para satisfacer sus

necesidades de comunicación. En Cuba muchas empresas cuentan con sistemas de información, sin embargo, tradicionalmente los datos son entregados en reportes impresos en papel y con el auge de las tecnologías éstos se necesitan en otros formatos, más fáciles de usar, y que puedan ofrecer un mayor nivel de detalle y flexibilidad, ya que las organizaciones se basan en información oportuna y precisa para gestionar las operaciones de negocios, identificar los riesgos, mejorar la productividad y aumentar la rentabilidad.

Una de las entidades afectadas por la necesidad de un avance en su sistema de información es la EICTV. Dicho centro, en su afán de alcanzar una mayor competitividad, busca la interacción de su sitio web con una herramienta capaz de generar y mostrar la información filtrada de la base de datos de forma rápida, precisa y confiable, pues la manera en que se obtiene hoy en día ocasiona deficiencias, como son la demora en la búsqueda de información en la fuente de datos, errores en el documento que conlleva a la poca confiabilidad del mismo y la pérdida de alguno de éstos aún sin perder total vigencia. Éstos desfavorables hechos están definidos como la **situación problemática**.

Por tanto, para resolver lo anterior se debe responder la siguiente interrogante, expuesta como **problema científico**:

¿Cómo garantizar la disponibilidad de la información filtrada que necesitan los directivos de la EICTV en su sistema virtual informativo?

Se define como **objeto de estudio** a los sistemas de gestión de información, teniendo como **objetivo general** desarrollar una herramienta que garantice la disponibilidad de los reportes en el sistema de información de la EICTV, donde el **campo de acción** de la investigación se centrará en los sistemas generadores de reportes.

Para desarrollar el objetivo de este trabajo se planifican las siguientes **tareas investigativas**:

- Descripción de varios generadores de reportes que existen actualmente.
- Selección de las diferentes tecnologías y lenguajes a utilizar para el desarrollo de **CreativeReport**.
- Análisis, diseño e implementación de **CreativeReport**.
- Creación de la estructura estándar del archivo XML.
- Definición de la DTD (definición del tipo de documento) por el cuál se va a regir nuestro archivo XML.
- Validación de las principales funcionalidades del software mediante pruebas de unidad para verificar su correcto desarrollo.

Realizar las actividades anteriores lleva todo un proceso investigativo del cuál se apoya de algunos métodos básicos como son el **analítico-sintético** que es indispensable para el análisis de las teorías, documentos, etc. Permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio y la **modelación** que se evidencia a través de los diagramas necesarios que se realizan.

El presente documento está estructurado en tres capítulos:

En el **Capítulo 1** se realiza un esbozo teórico sobre generadores de reportes y las principales tendencias y tecnologías que sirven de soporte para el trabajo desarrollado.

En el **Capítulo 2** se realiza el análisis, diseño e implementación de **CreativeReport**, se crea la estructura estándar del archivo XML y se procede a definir la DTD (definición del tipo de documento) por el cuál se va a regir nuestro archivo XML.

En el **Capítulo 3** se exponen las pruebas de unidad realizadas a las principales funcionalidades del producto para verificar su robustez.

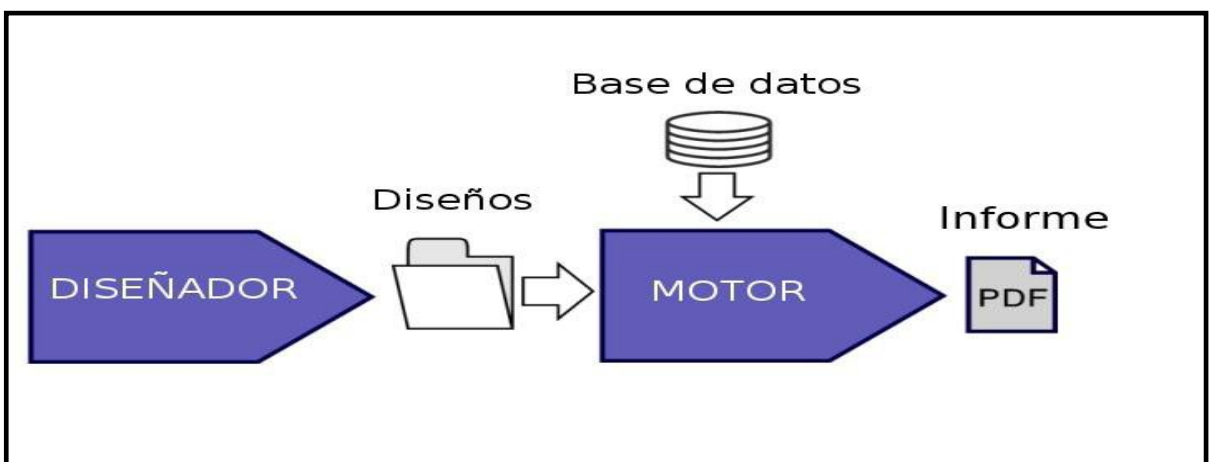
# Capítulo 1

## Fundamentación Teórica

El presente capítulo se centrará en la descripción de varios sistemas que gestionan reportes mencionando algunas de sus principales características y funcionalidades, además de nombrar varias la tecnología, herramientas y lenguaje propuestos para el desarrollo de CreativeReport, destacando también algunas definiciones que se estimaron relevantes para un mejor entendimiento del lector.

### 1.1 Rasgos generales del funcionamiento de generadores de reportes.

Los generadores de reportes de manera estándar se caracterizan por estar compuestos por dos elementos básicos, un diseñador ó editor de informes y un motor que los genere.



De acuerdo al esquema de la figura anterior las principales actividades o tareas que debe realizar un generador de reportes son las siguientes:

- Conformar el diseño de los reportes.
- Luego de obtener los datos se genera en el formato requerido el informe final de acuerdo a lo diseñado previamente.

## **1.2 Diseño de informes**

El diseño de un reporte o informe no es más que una plantilla que modela y encierra las características con las cuáles quiere que se genere el reporte deseado. De esta manera una vez que se insertan todos los componentes requeridos en tiempo de diseño en las distintas secciones, persistirán en la plantilla estas características y componentes, que a la hora de generar el reporte aparecerá como se diseñó, con la diferencia de que estarán todos los datos volcados en su interior.

## **1.3 Características de varios generadores de reportes.**

Son aplicaciones informáticas que más comúnmente encontramos en nuestros días, en las cuáles se expone una interfaz que visualiza la información obtenida de una base de datos, con la particularidad de que éstos se vuelcan en la apariencia que fue previamente diseñada. Es por ello que la generación de informes es el principal punto para la toma de decisiones sobre la producción por parte del personal encargado. Resulta necesario realizar un estudio de éstas aplicaciones, con el objetivo de extraer formas de solución, mecanismos y resultados que se puedan usar en la realización del presente trabajo.

### **1.3.1 JasperReports**

Es una herramienta de creación de informes Java open source que tiene la habilidad de entregar contenido enriquecido en el monitor, a la impresora, o a ficheros PDF, HTML, XLS, CSV y XML.

Está escrito completamente en Java y puede ser usado en gran variedad de aplicaciones de Java, incluyendo J2EE ó aplicaciones web, para generar contenido dinámico. Su propósito principal es ayudar a crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible.

JasperReports se usa comúnmente con iReport, unos front-end gráficos open source para la edición de informes.

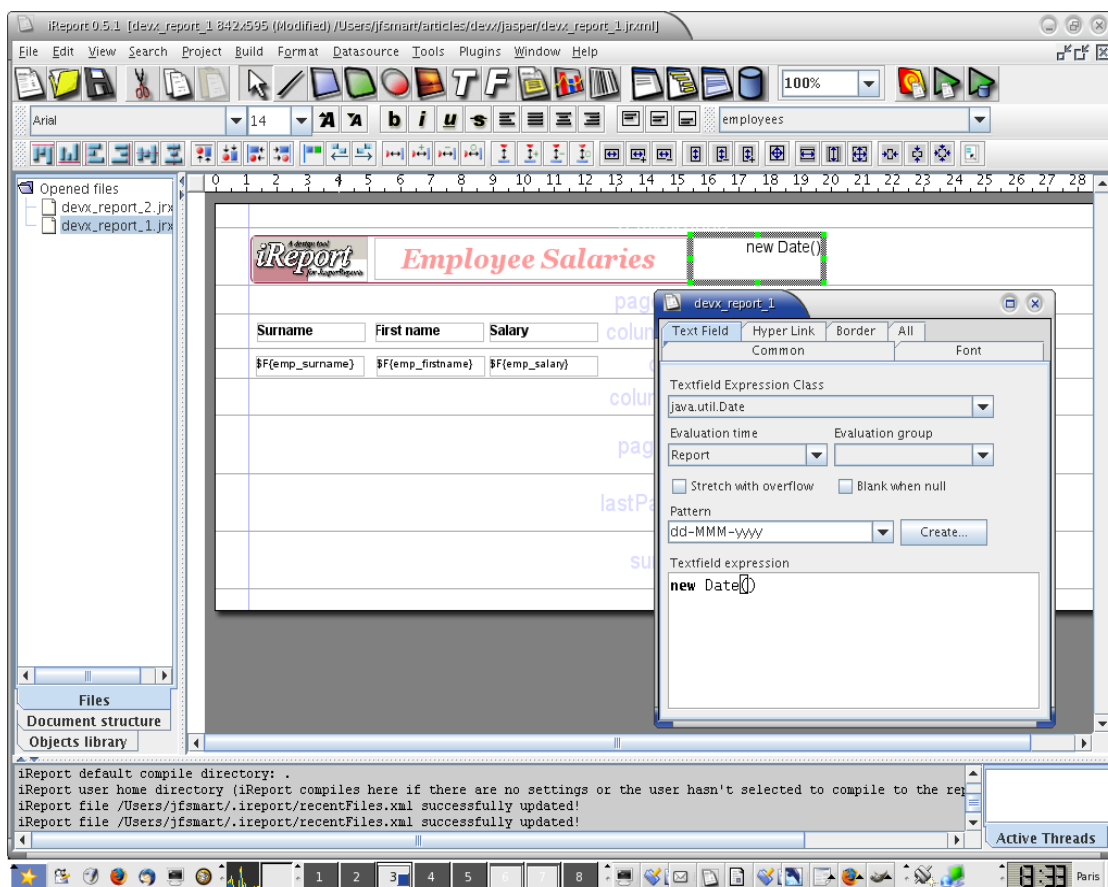


Fig. 2: Interfaz visual de JasperReports.

### 1.3.2 NCRReport

Es un generador de informes que a su vez permite el diseño visual de las plantillas a generar. Tiene una apariencia visual amigable permitiendo, desde el menú principal, la utilización de los distintos componentes en la paleta de herramienta. Ejemplos de los componentes de los cuáles dispone están campo de expresión, número de página, texto y etiquetas, imágenes, hipervínculo y autoforma (línea, cuadro) y tabla de referencia cruzada.(NCREPORT).

Los reportes diseñados son almacenados en archivos con formato XML permitiendo la lectura del mismo por el motor generador. Esta implementado

sobre C++ y Qt y bajo licencia GPL, lo cual permite su uso y modificación en caso de interés para cualquier persona ó entidad.

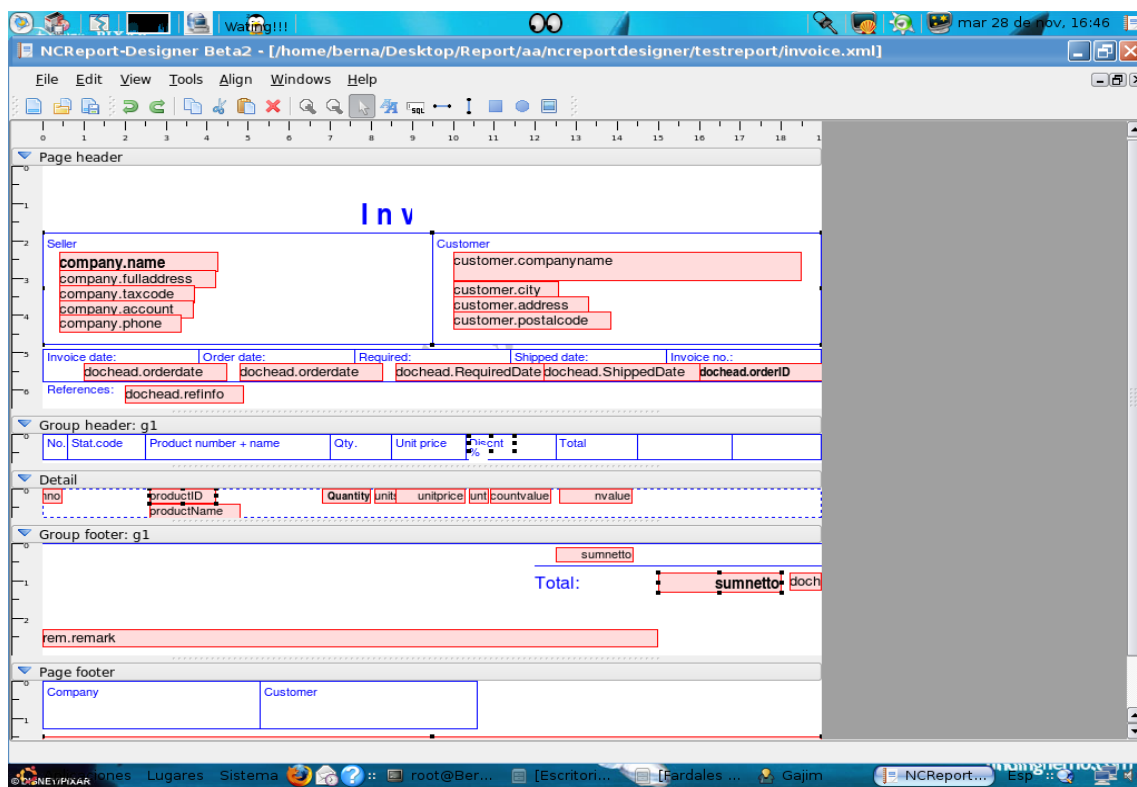


Fig. 3 Ambiente de edición del NCReport.

### 1.3.3 Ruport

Es un sencillo sistema de información extensible que ofrece las estructuras de datos estándar que le permite comparar fácilmente los datos de cualquier número de fuentes, incluyendo archivos CSV, ActiveRecord, respaldo de bases de datos, y SQL en bases de datos en bruto, incluso legado. Además de proporcionar un sistema de formato muy potente, que sienta las bases para PDF, CSV, HTML y de texto basado en los informes.

En el aspecto de la ayuda y mantenimiento ofrece un paquete de servicios públicos que se llama Ruport-Útil que proporciona soporte para una amplia gama de funciones de nivel superior, como la generación de la factura, informe de correo, gráficas, generación de código, y la producción adicional formatos.



### **1.3.4 CrystalReports**

CrystalReports es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos.

Varias aplicaciones, como Microsoft Visual Studio, incluyen una versión OEM de Crystal Reports como una herramienta de propósito general de informes/reportes. Se convirtió en el escritor de informes estándar cuando Microsoft lo liberó con Visual Basic.

Al instalarlo en un equipo se utiliza para seleccionar filas y columnas específicas de una tabla de datos compatibles, pueden organizar los datos en el informe en el formato que necesiten. Una vez que el diseño está completo, el informe se puede guardar/salvar como un archivo con extensión rpt. Se puede acceder nuevamente al informe reabriendo el mismo, y poder refrescar los datos. Si la base de datos se ha actualizado, el informe va a reflejar estas actualizaciones.

El reporte puede ser visto de antemano en la pantalla (con el comando CTRL+R), imprimir en papel, ó se exportado a uno de los diversos formatos de archivo soportados: PDF, rtf de Microsoft Word, .xls de Microsoft Excel (en dos versiones, una de sólo datos y la otra convencional), .CSV o texto (.txt).

### **1.3.5 PHPReports**

Es una biblioteca soportada solamente por el lenguaje de php. Es un generador de reportes de código abierto sin interfaz visual con soporte para múltiples gestores de bases de datos. El formato que usa para sus informes es el XML (un metalenguaje muy flexible), que a pesar de usarlo ha quedado prácticamente obsoleto por su complejidad y poca flexibilidad en el código.

## **1.4 Principales tendencias.**

El número de ordenadores está aumentando progresivamente en las organizaciones, pero este fenómeno ocasiona otra serie de problemas que la empresa necesita solucionar, como son enviar la información correcta a la persona indicada, actualizar los programas que se están usando, controlar el acceso a los datos, etc. Cuando se emplea una tecnología que cambia tan rápidamente sin disponer de los controles adecuados, el riesgo más inminente es que el sistema quede obsoleto. Es por ello que cada día se construyen nuevo software capaz de satisfacer las exigencias de las entidades, y se dedique tiempo a pensar en el uso o no de algunos lenguajes y tecnologías que le ayudarán a implementar un producto, con el objetivo de encaminarse por uno de ellos.

Al haber realizado un estudio de las tendencias que existen en la actualidad al desarrollar un generador de reportes, se seleccionan algunas de ellas que cumplen de forma equitativa los conceptos más importantes para los usuarios de la EICTV según sus condiciones y requisitos. Dentro de esta selección se destaca como metodología de desarrollo de software el RUP (Rational Unified Process), ya que su objetivo es producir software de alta calidad y cumplir con los requerimientos de los usuarios dentro de una planificación establecida; además de que es un proceso dirigido por *casos de uso*, donde un caso de uso es *una* especificación de una secuencia de acciones que el sistema puede llevar a cabo *y es iterativo e incremental*. Tiene la característica de tomar en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software, que algunas de ellas son manejar los requisitos, modelar el software visualmente, verificar la robustez del producto, entre otras; que son imprescindibles para CreativeReport.

El lenguaje de programación establecido es php, por ser multiplataforma. Está orientado completamente al desarrollo de aplicaciones web dinámicas con

acceso a información almacenada en una base de datos; además de que el código fuente escrito es invisible al navegador y al cliente, ya que es el servidor (en este caso el Apache v2.2, es el que está utilizando la EICTV actualmente) el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en éste lenguaje sea segura y confiable.

El gestor de bases de datos seleccionado es Mysql, debido a que se utiliza en la EICTV, y modelamos los diagramas correspondientes de ingeniería de software, en función de nuestros procedimientos a través de Visual Paradigm, que es una herramienta para UML profesional que permite dibujar de manera eficaz todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación; además de que ayuda a una rápida construcción de aplicaciones con mejor calidad.

## **1.5 Conclusiones**

Es evidente que con el auge de las tecnologías, los generadores de reportes deberán responder a las necesidades del cliente de manera más ágil, estar desarrollados con la última generación de herramientas y sin ocupar lo menos posible el procesador de la computadora, además de que deberán adaptarse a las características de los sistemas web de información.

Las tendencias que existen en la actualidad para la implementación de un generador de reportes son muy variadas, solo depende de las necesidades específicas de cada cliente, por tanto, a partir de haber estudiado algunos de ellos con sus respectivas funcionalidades se llega a la conclusión de que ninguno garantiza el cumplimiento de los requisitos del cliente (EICTV) para su uso.

En el próximo capítulo se ofrece una propuesta de un típico generador de reportes para solucionar las deficiencias del sistema de la EICTV.

## Capítulo 2

### Construcción de la solución propuesta

En este capítulo se presentan las principales clases y funcionalidades de nuestro software **CreativeReport** a través de la modelación y el diseño de algunos componentes relevantes, no sin excluir las menos importantes; además de desarrollar la implementación del mismo, que no es más que mostrar los resultados finales de la herramienta.

#### 2.1 El sistema virtual de la EICTV en la actualidad.

El sistema virtual de la EICTV actualmente está activo, trabaja con 5 módulos, ellos son administración, tecnología, eventos y festivales, docencia y archivo fílmico; cada uno de ellos gestiona su información de forma independiente, pero constituye un solo módulo integrado.

Los jefes de cada uno de los departamentos no encuentran en el sitio la información adecuada y precisa que necesitan, para ellos el sistema virtual no proporciona información de valor que ayude a la toma de decisiones. Estos usuarios de alguna manera necesitan de información filtrada desde la base de datos, y realizan toda una serie de actividades para abastecerse de la misma. Estas acciones se definen como el **proceso obtener informe**.

##### 2.1.1 Modelo de negocio

Un modelo del negocio es un modelo que organiza y presenta los detalles importantes de los problemas reales que se vinculan con un posible sistema

informático a construir. Comprende la estructura dinámica de la organización en la cual se va a implementar un sistema, los problemas actuales de esta organización e identifica las mejoras potenciales. Asegura que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización. Del modelo del negocio se derivan los requerimientos del sistema que va a soportar la organización.

### 2.1.2 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término *actor* significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor.

Actores del negocio	Justificación
Jefe	Es la persona que solicita un determinado informe y se abastece de la información del mismo.

**Tabla 1.** Actores del negocio.

### 2.1.3 Trabajadores del negocio

Un trabajador define el comportamiento y las responsabilidades de un individuo que actúa en el negocio realizando una o varias operaciones, interactuando con otros trabajadores del negocio y manipulando entidades del negocio.

A continuación se muestran los trabajadores del negocio con su correspondiente justificación.

Trabajadores del negocio	Justificación
trabajador	Es el responsable de realizar el informe, y garantizar que el mismo esté correctamente desarrollado. Además de comprobar que el directivo que solicite el informe tenga acceso a la misma.

**Tabla 2.** Trabajadores del negocio.

#### **2.1.4 Reglas del Negocio**

Las reglas del negocio describen políticas que deben cumplirse ó condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio, además determinan y limitan como fluye la información a través del sistema. Existen varios tipos de reglas del negocio, las que se estipulan en el proceso anteriormente detallado en la tabla son las siguientes:

##### *2.1.2.1 Reglas de acción*

- Un informe solo podrá ser solicitado al trabajador por un Jefe.
- El jefe deberá solicitar solamente información de su departamento, a excepción de la máxima dirección de la entidad.

#### **2.1.5 Modelo de casos de uso del negocio**

El modelo de casos de uso del negocio es un modelo que describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes, es decir, describe las funciones que el negocio pretende realizar y su objetivo básico es describir cómo el negocio es utilizado por sus clientes y socios.

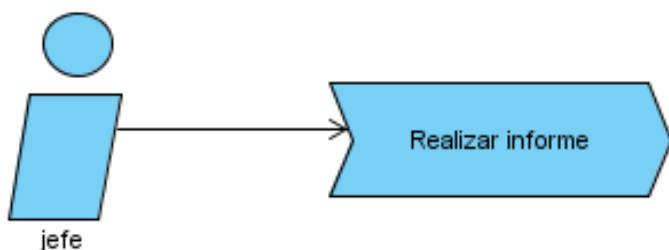


Fig. 4: Modelo de casos de uso del negocio.

#### 2.1.6: Descripción textual del caso de uso del negocio *Realizar informe*

nombre del caso de uso del negocio		Realizar informe
Actor(es) del negocio:		Jefe (inicia), trabajador
propósito:	El directivo recibe informe con los datos precisos que necesita para la toma de decisiones de su departamento.	
resumen:  Este caso de uso comienza cuando el jefe emite la orden de que necesita un informe con datos específicos, el trabajador recibe las características del informe y procede a su realización.		
Casos de uso asociados		----
flujo de trabajo alternativo		
acción del actor		respuesta del negocio
1. El caso de uso inicia cuando el jefe emite una orden al trabajador de realizar un informe.		2. El trabajador recibe la orden emitida.

	3. Verifica que el jefe esté solicitando información de su departamento. En caso de que no tenga autorización (4) y (5). Si tiene acceso la acción (6).
	4. Envía notificación de no acceso a la información.
5. Recibe la notificación (termina el caso de uso).	
	6. El trabajador procede a realizar el informe accediendo a la base de datos.
	7. Archiva una copia del documento realizado actualizando el archivo de informes.
	8. Entrega el informe solicitado.
9. Recibe el informe realizado.	
10. Termina el caso de uso.	

**Tabla 3:** Descripción textual del caso de uso del negocio *Realizar informe*, donde se evidencia la actividad del actor (Jefe) y la respuesta del negocio.

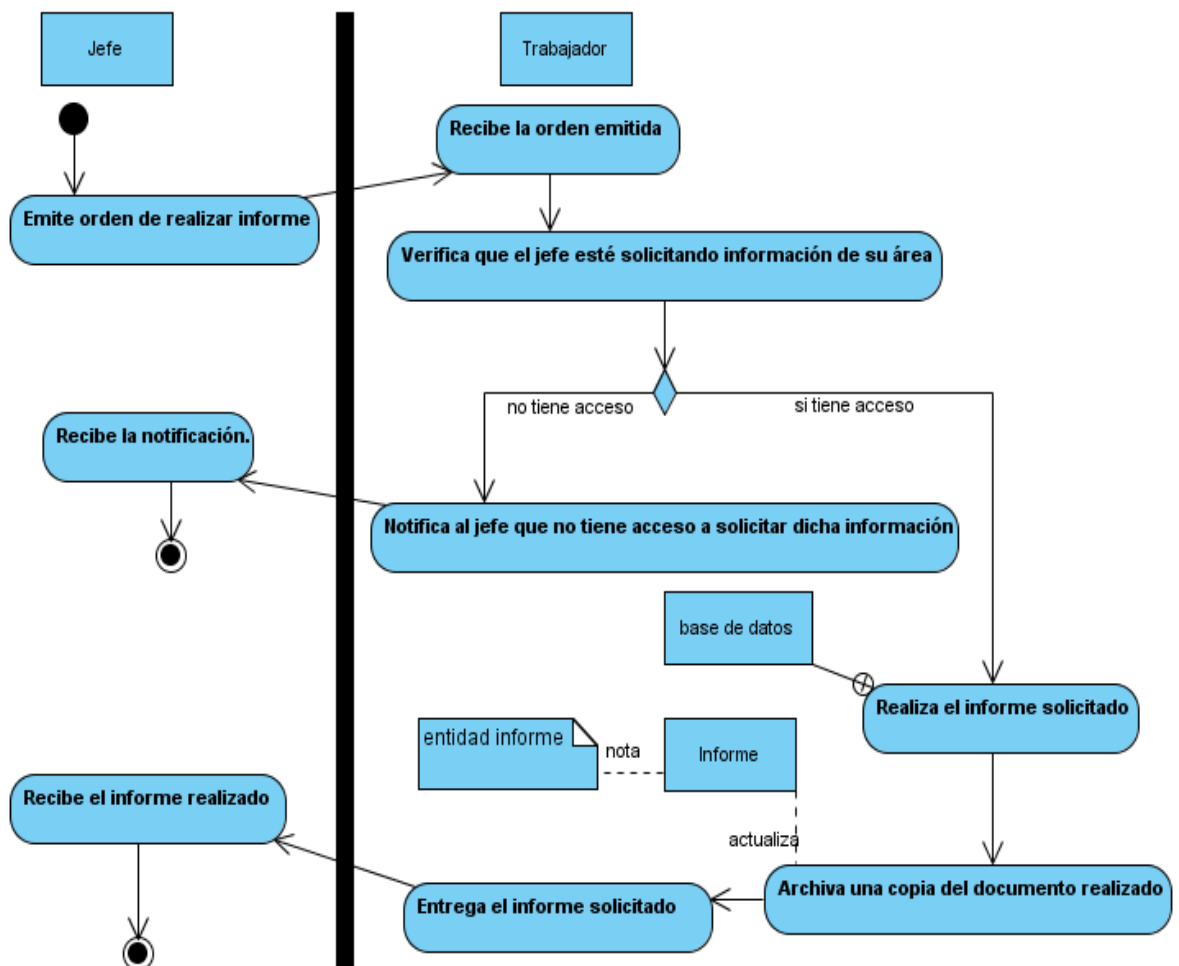
En esta tabla se identifican todos los pasos que componen al proceso de negocio *obtener informe*, se muestran todas las actividades que el actor jefe y la respuesta del negocio que es la acción del trabajador realizan para complementar esta tarea.

### 2.1.7 Diagrama de actividades



Un diagrama de actividades es una estructura que describe gráficamente la secuencia de actividades que, en conjunto producen algo para el actor. El diagrama de actividades es un grafo de actividades que contienen estados en que puede hallarse una actividad. Describe un proceso que explora el orden de las tareas o actividades que logran los objetivos del negocio.

En el siguiente *diagrama de actividades* se aprecia con más detalle cómo fluye el caso de uso del negocio *Realizar informe* y la interacción de los actores del negocio implicados.



**Fig. 5** Diagrama de actividades del caso de uso del negocio *Realizar informe*.

Se decide automatizar el caso de uso del negocio *Realizar informe*. Después de conocer cómo fluyen sus actividades y bajo qué restricciones, veamos la propuesta que se ofrece en dependencia de los requisitos que desea la EICTV.

#### **2.1.8 Requisitos *funcionales***

Los *requisitos funcionales* son capacidades o condiciones que el sistema debe cumplir

##### **Requisitos funcionales:**

1. Mostrar reportes de información filtrada.

#### **2.1.9 Requisitos no *funcionales***

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

A continuación se detallan los requerimientos no funcionales del sistema:

##### **Requisitos no funcionales:**

###### *Requisitos de software:*

1. Sistema operativo Windows 95 ó superior.

###### *Requisitos de hardware:*

1. Disponer de capacidad de disco duro 20 GB mínimo.

*Requisitos de seguridad:*

1. La información manejada por el sistema está protegida de acceso no autorizado.

## 2.2 Propuesta

**CreativeReport** es una herramienta para reportes que muestra informes en el sistema virtual de información de la EICTV a partir de la lectura de un archivo con formato XML.

Está implementado en el editor de código NetBeans v6.8, y no se rige por ningún estándar definido para realizar un software de estas cualidades, sino que de los existentes estudiados en el capítulo anterior, tomamos las funcionalidades más relevantes de acuerdo a las condiciones y necesidades de uso de los usuarios directivos.

- **JasperReports:** Se reutiliza su flexibilidad y simpleza para imprimir los reportes a otros formatos como PDF y Excel.
- **NCRReport:** La principal característica de este generador es que guarda sus reportes en archivos con formato XML, con la cual ofrece mayor disponibilidad de entendimiento y transportación de la información. Esta ventaja se incluye también en nuestro producto.
- **PHPReports:** Librería con soporte para múltiples gestores de bases de datos, es la función principal de este generador, que se le adjunta también a nuestro con el objetivo de que sea lo más dinámico posible, es decir, en caso de cualquier cambio del gestor de base de datos se realizan algunas modificaciones mínimas de manera tal que **CreativeReport** no quede obsoleto.

**CreativeReport** es capaz de satisfacer los requisitos anteriores tanto funcionales como no funcionales. Mostrará reportes en un tiempo casi

automático debido a la simplicidad que aporta el uso de guardar la información en archivos con formato XML; que se ha desarrollado rápidamente y no es menos cierto que una gran cantidad de desarrolladores han adoptado este estándar, tan importante para la web como lo es y lo fue HTML. Ahora resulta que es la herramienta más común para la transmisión de datos entre todas las aplicaciones.

### **2.3 Métodos de validación para XML**

Existen varios métodos de validación que comprueban que el documento XML esté bien formado (*Well Formed*), entre ellos está el DTD que es una definición en el documento XML, que determina restricciones en la estructura y sintaxis del mismo. Se puede incluir dentro del archivo, pero normalmente se almacena en un fichero de texto separado. Se emplea para determinar la estructura mediante etiquetas (*tags*), y es el más sencillo para validar, pero presenta la limitación de no permitir nuevas ampliaciones del XML. Esta desventaja no nos afecta, pues la estructura de nuestro XML no necesita de nuevas ampliaciones, entiéndase por esto, que no será necesario modificarlo.

El DTD definido para el software es el que veremos a continuación, podemos observar los elementos que va a contener el título del reporte, que es una obligación definirlo; otro elemento es la descripción que es opcional, depende del usuario si la va a utilizar; el elemento bloque, que va a constituir la tabla con los datos que serán extraídos de la base de datos a través de una consulta; y luego podemos insertar una hoja de estilo, que no es más que una plantilla que contendrá las clases que regirán el estilo de cada elemento mencionado.

```

<?xml version='1.0' encoding='UTF-8'?>

<!-- Reporte -->
<!ELEMENT report (title,description?,block+,styles)>
<!ATTLIST report
    stylesheet CDATA #REQUIRED
>

<!-- Título del reporte -->
<!ELEMENT title (#PCDATA)>

<!-- Breve descripción del reporte -->
<!ELEMENT description (#PCDATA)>

<!-- Bloques o Grupos dentro del reporte -->
<!ELEMENT block (subtitle?, sql+, table, summary?, styles-block?)>

<!-- Título del bloque o grupo -->
<!ELEMENT subtitle (#PCDATA)>

<!-- Consulta SQL -->
<!ELEMENT sql (query, where?, group?, order? )>
<!ATTLIST sql
    id CDATA #REQUIRED
>

<!-- Declaración de la consulta (SELECT) -->
<!ELEMENT query (#PCDATA)>
<!ATTLIST query
    limit CDATA #IMPLIED
>

```

```

<!-- Cláusula WHERE -->
<!ELEMENT where (#PCDATA)>

<!-- Cláusula GROUP BY -->
<!ELEMENT group (#PCDATA)>

<!-- Cláusula ORDER BY -->
<!ELEMENT order (#PCDATA)>

<!-- Grupo de columnas de la tabla -->
<!ELEMENT table (column+)>
<!ATTLIST table
    sql-id CDATA #REQUIRED
>

<!-- Columna de la tabla -->
<!ELEMENT column (title)>
<!ATTLIST column
    style CDATA #IMPLIED
    field CDATA #REQUIRED
    id CDATA #IMPLIED
>

<!-- Sumario de la tabla (totales) -->
<!ELEMENT summary (column+)>
<!ATTLIST summary
    sql-id CDATA #REQUIRED
>

<!-- Estilos CSS a aplicar al bloque o grupo -->
<!ELEMENT styles-block (subtitle-style, table-header-style, table-row-style, table-alternate-style, summary-style)?>

```

```

<!-- Estilo del subtítulo -->
<!ELEMENT subtitle-style (#PCDATA)>
<!ATTLIST subtitle-style
    style CDATA #IMPLIED
    class CDATA #IMPLIED
>

<!-- Estilo del encabezado de la tabla -->
<!ELEMENT table-header-style (#PCDATA)>
<!ATTLIST table-header-style
    style CDATA #IMPLIED
    class CDATA #IMPLIED
>

<!-- Estilo de las filas -->
<!ELEMENT table-row-style (#PCDATA)>
<!ATTLIST table-row-style
    style CDATA #IMPLIED
    class CDATA #IMPLIED
>

<!-- Estilo de las filas alternas -->
<!ELEMENT table-alternate-style (#PCDATA)>
<!ATTLIST table-alternate-style
    style CDATA #IMPLIED
    class CDATA #IMPLIED
>

```

**Fig. 6:** DTD definido para CreativeReport por el cual será analizado el archivo XML.

Una vez que hemos analizado el DTD que valida el archivo con formato XML, realizamos la estructura de dicho archivo. Es muy sencilla y flexible de código, tanto, que un usuario con pocos conocimientos informáticos podría entender la definición. Ésta precisamente es una de las ventajas que proporciona usar este metalenguaje, pues se define con el lenguaje del desarrollador.

La definición de la estructura es la siguiente:



```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE report SYSTEM "reporte.dtd">
<report stylesheet="">
  <title>
  </title>

  <description>
  </description>

  <block>
    <subtitle>
    </subtitle>

    <sql id="">
      <query limit="">
      </query>
      <where>
      </where>
      <group>
      </group>
      <order>
      </order>
    </sql>

    <table sql-id="">
      <column id="" field="" style="">
        <title>
        </title>
      </column>
    </table>

```

```

    <summary sql-id="">
      <column id="" field="" style="">
        <title>
        </title>
      </column>
    </summary>

    <styles-block>
      <subtitle-style class="" style="" />
      <table-header-style class="" style="" />
      <table-row-style class="" style="" />
      <table-alternate-style class="" style="" />
      <summary-style class="" style="" />
    </styles-block>

  </block>

  <styles>
    <title-style class="" style="" />
    <description-style class="" style="" />
    <subtitle-style class="" style="" />
    <table-header-style class="" style="" />
    <table-row-style class="" style="" />
    <table-alternate-style class="" style="" />
    <summary-style class="" style="" />
  </styles>
</report>

```

**Fig. 7:** Estructura del archivo XML donde será guardada la información para generar el reporte deseado.

La estructura del archivo XML es sencilla, pero capaz de soportar cualquier información. Está compuesto por bloques de código que van a permitir fundamentalmente como mínimo una consulta, que es la que especifica los datos que se van a volcar dentro del reporte a la hora de mostrarlos. Admite de manera diversa todas las consultas a la base de datos que el usuario desee, o sea, permite cualquier variante de diseño al reporte y de consulta. Está previsto para la creatividad del usuario.

## 2.4 Modelo del sistema

### 2.4.1 Actores del sistema

Un actor es una idealización de una persona externa, de un proceso, o de una persona que interactúa con un sistema, un subsistema o una clase. Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema, puede representar el rol que juega una o varias personas, un equipo o un sistema automatizado. Suelen ser definidos en jerarquía de generalización, en las cuáles una descripción abstracta del actor es compartida y aumentada por una ó más descripciones específicas del actor.

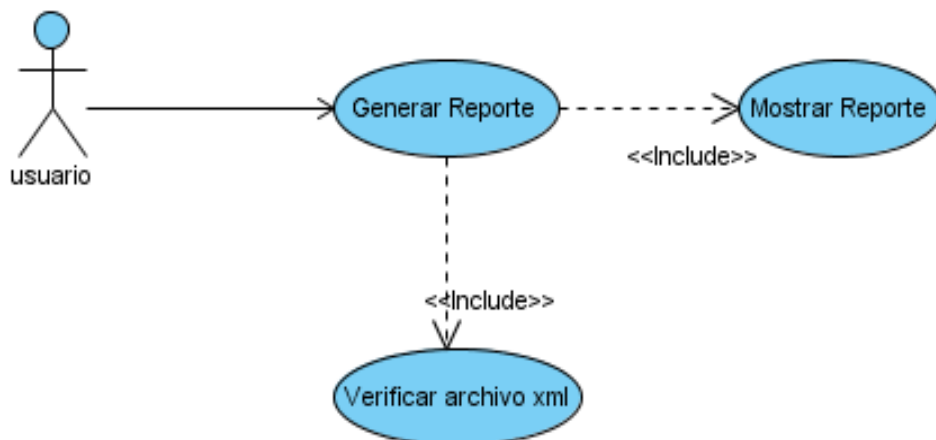
Por lo general estimula el sistema con eventos de entradas o recibe algo de él, o sea, es un rol de un usuario, que puede intercambiar información ó puede ser un recipiente pasivo de información y representa a un ser humano, a un software, ó a una máquina que interactúa con el sistema.

Un actor es una entidad externa del sistema que de alguna manera participa en la historia del caso de uso.

Actores /función asociada	
Actores	Función
Usuario (en este caso el directivo).	Es aquél que tiene el acceso a ver los reportes desde su sesión de usuario.

**Tabla 4.** Descripción de actores del sistema.

## 2.4.2 Diagrama de casos de uso del sistema.



**Fig. 8:** Diagrama de casos de uso del sistema.

Nuestro modelo está compuesto por tres casos de uso, uno de ellos es *generar reporte* que tratará todas las actividades básicas que tengan que ver con la creación del reporte y mostrarlo; pero para completar este caso de uso incluimos dos casos de usos imprescindibles. Dichos casos dependen de generar reporte, que es el que los contiene. Fueron divididos en dos porque cada uno se enfoca en una secuencia de tareas distintas; donde el caso de uso *verificar archivo XML* rige todas aquellas funcionalidades que traten con tal archivo, entendiéndose por esto validarlo, leer la información que contiene y garantizar que esté bien formado, o sea, bien estructurado. El otro caso de uso incluido *Mostrar Reporte* administra todas las salidas que pueda mostrar el sistema incluyendo el reporte, centrándose específicamente en los mensajes de error.

## 2.5 Modelo de análisis

El modelo del análisis define una estructura que es una entrada esencial para modelar el sistema, incluyendo la creación del modelo de diseño. Es un modelo conceptual que permite obtener un bosquejo del diseño del sistema, incluyendo su arquitectura. El análisis forma parte del proceso de desarrollo de software. Su objetivo fundamental consiste en formular el modelo del dominio del problema durante el análisis, se analizan los requerimientos que fueron

descritos en la captura de requisitos. Los mismos son refinados y estructurados con el propósito de adquirir una comprensión más precisa y una descripción que sea fácil de mantener, siendo de ayuda para la estructuración de la aplicación íntegra. Ello proporciona una estructura centrada en el mantenimiento, basada en aspectos tales como la flexibilidad ante los cambios y la reutilización.

### **2.5.1 Modelo de clases del análisis.**

El diagrama de clases del análisis representa básicamente los conceptos del dominio del problema. Representa los aspectos del mundo real, no de la implementación automatizada de dichos aspectos. En este diagrama se excluyen las características irrelevantes y se utilizan los nombres existentes en el dominio, sin agregar aspectos inexistentes.

En el modelo de clases del análisis se definen varios artefactos, tal es el caso de las clases de interfaz, entidad y control. La disposición de clases del sistema se especifica con estructuras de herencia y relaciones entre clases. El diagrama que se desarrolla durante el análisis persigue la solución ideal.

Una clase del análisis representa una abstracción de una o varias clases y/o subsistemas del modelo del diseño.

### **2.5.2 Descripción de clases del análisis**

Para facilitar el análisis, se clasifican las clases en:

#### **Clase de control**

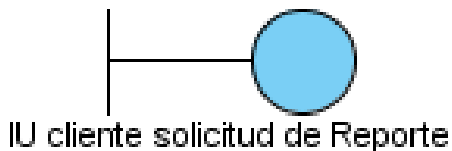
- Contienen la lógica del negocio y manejan la mayoría de los algoritmos y operaciones que se realizan en el sistema.
- Representan coordinación, secuencia, transacciones, y control de otros objetos.
- Se usan mucho para encapsular el control de un caso de uso concreto.
- Manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos (de interfaz y de entidad).



**Fig.9** clase control.

### **Clase de interfaz**

- Estas clases contienen los componentes necesarios para construir las vistas que se utilizan para la entrada de datos y para mostrar los resultados.
- Los atributos de estas clases son públicos de manera tal que otras clases puedan acceder a ellos en cualquier momento.
- Modelan la interacción entre el sistema y sus actores (usuarios y sistemas externos).
- Reúnen los requisitos en los límites del sistema.
- Reciben y presentan información y peticiones de y hacia los actores.



**Fig.10** clase interfaz.

### **Clase de entidad**

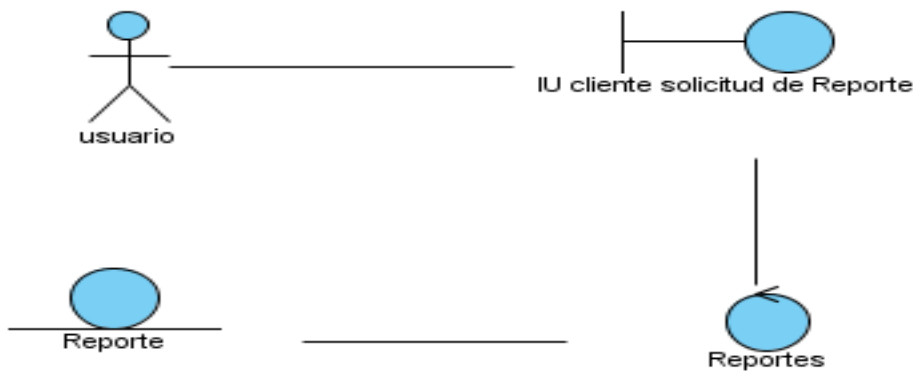
- Estas clases contienen la información necesaria del sistema en cuestión, la mayoría realizan operaciones simples sobre los datos que poseen.
- Modelan información que posee una larga vida.
- Modelan la información y el comportamiento asociado a algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.
- Derivan normalmente de una clase de entidad del negocio.

- Aísla los cambios en la información que representa.



**Fig.11** clase entidad.

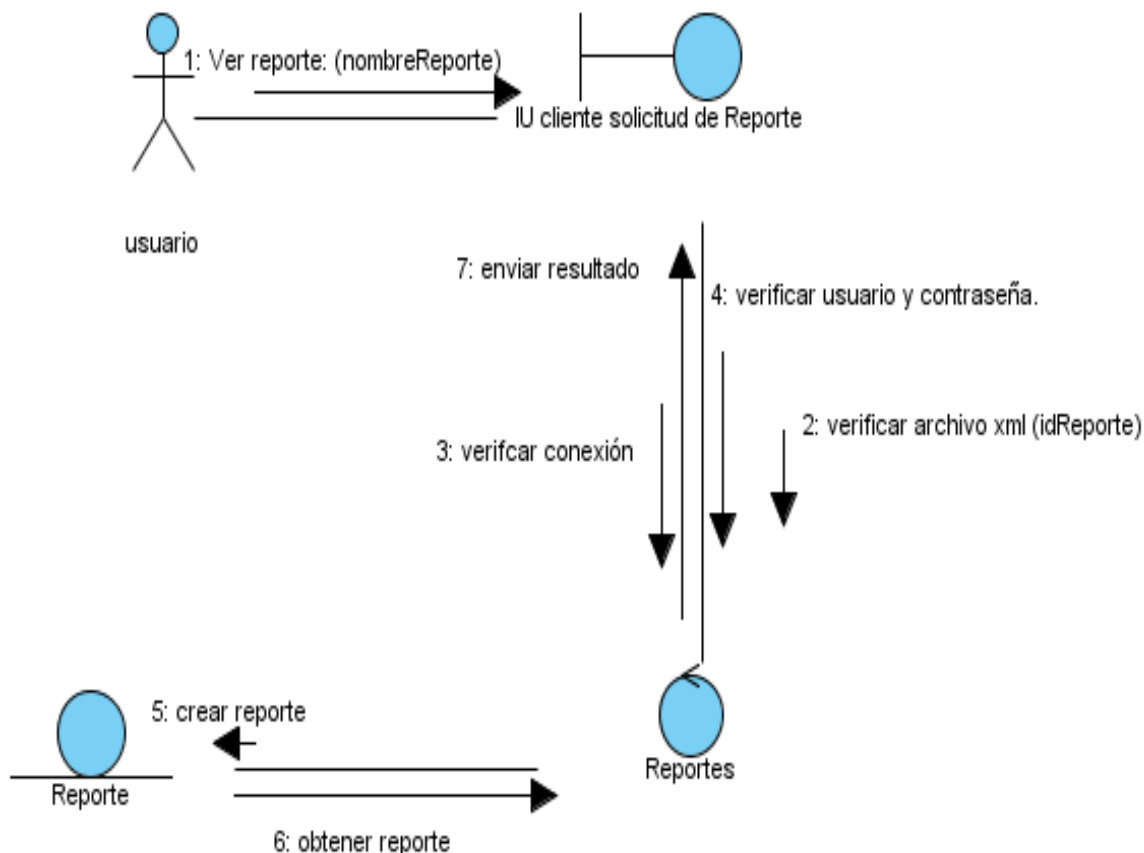
### 2.5.3 Diagrama de análisis del sistema.



**Fig. 9:** Diagrama que muestra las relaciones entre las diferentes clases.

Aclarar que aunque este trabajo no trata la implementación de la interfaz visual, se identifica aquí clase interfaz *cliente solicitud de reporte* enfocados a una vez que se complemente nuestra herramienta con el sitio informativo de la escuela, ya en ese caso que estará trabajado por sesiones, existe la sesión de usuario que es la que se representa como *IU cliente solicitud de reporte* que se conectará con nuestra herramienta y realizar todas las tareas pertinentes para mostrarle la información, en nuestro caso, al directivo.

A continuación mostraremos el mismo diagrama pero con las acciones que realizan para mostrar el reporte.



**Fig. 10:** Diagrama que describe el caso de uso del sistema Generar Reporte.

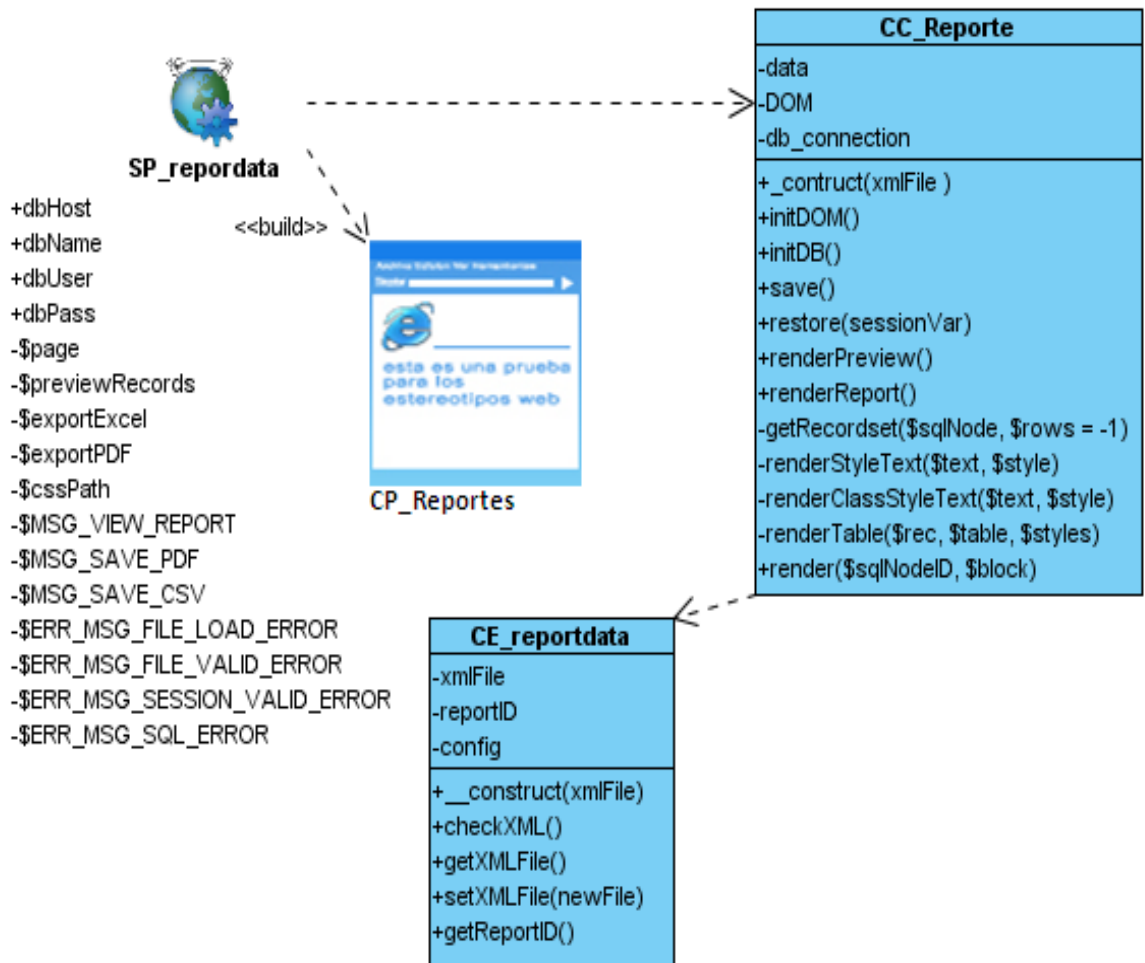
Este diagrama está más completo, pues muestra de manera consecutiva todas las acciones que realizará el sistema, desde pedir el XML, hasta el resultado final, siempre enfocándose a una vez que ya esté trabajando conjuntamente con la intranet virtual de la EICTV.

El usuario comenzará por desear ver el reporte, como es basado en sesiones el sistema de la escuela y que por esta característica no guarda ninguna variable, nuestro producto identificará el id del reporte que se solicita y buscará el archivo XML que se identifique con el código único del reporte solicitado, que a su vez contendrá la información requerida, generándola al instante. Tendrá éxito solamente a la muestra de la información aquél usuario que después de haberse identificado, tenga acceso a solicitar un reporte.

## 2.6 Modelo de diseño

Para el desarrollo de este flujo de trabajo, se tiene como premisa fundamental la modelación del sistema, estructurando la información de forma tal, que cumpla con los requerimientos especificados por el cliente. Se identifican los objetos pertinentes, se definen interfaces y establecen relaciones entre ellos. Luego se organiza la arquitectura del sistema de manera que contemple las formas de acceder y procesar las informaciones que éste maneje.

### 2.6.1 Diagrama de diseño del sistema.



**Fig.11** Diagrama de diseño del sistema

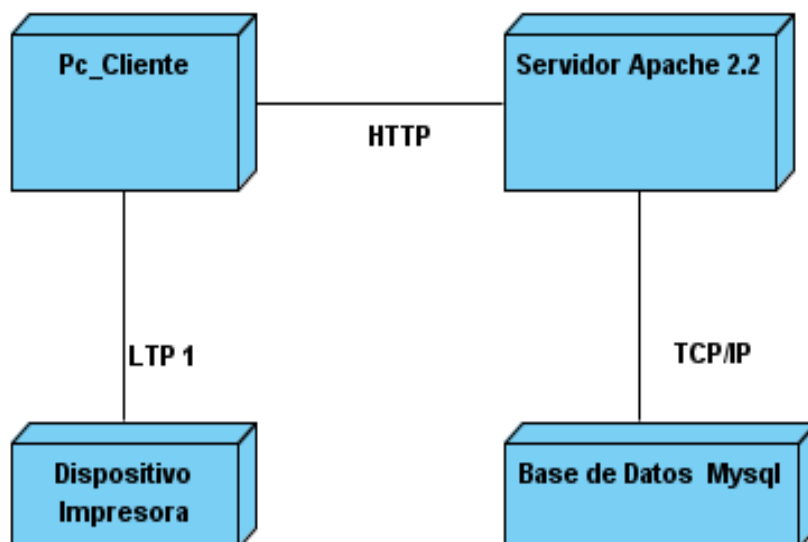


En este diagrama se muestran todas las funcionalidades que componen a CreativeReport en el caso de las clases *CE\_repordata* y *CC\_reporte*. De manera general dicho diagrama describe las conexiones que existen entre la página servidora, la página cliente y las clases mencionadas anteriormente. En este caso se comunican de la siguiente manera; la página servidora *SP\_reportdata* requiere de la validación de conexión del usuario, ésta a su vez al validar el acceso, pasa a construir el archivo XML necesitando del contenido físico, o sea, de los datos del XML, “devolviéndole” el resultado al navegador donde interpreta el código y posteriormente lanza el reporte generado en la *CP\_Reportes* que es la página del cliente.

## **2.7 Diagrama de despliegue**

El diagrama de despliegue permite apreciar de forma visual como se encuentran relacionados físicamente los componentes de la aplicación. Describe la distribución física del sistema, muestra cómo están distribuidos los componentes de software entre los distintos nodos de cómputo. Este diagrama permite comprender la correspondencia entre ambas arquitecturas software y hardware. Los nodos se utilizan para modelar la topología del hardware sobre el cual se ejecuta el sistema. Representa típicamente un procesador o un dispositivo sobre el que se pueden desplegar los componentes. En nuestro caso, se muestran todos los activos físicos que deben estar presentes para implantar CreativeReport, pues se necesita mínimo de una computadora que a través del protocolo HTTP se conecte al servidor, donde a su vez éste se comunicará con el gestor de bases de datos que contiene.

La conexión de la impresora está disponible porque se necesita de la impresión de los reportes, ya que una de las características de nuestro software es precisamente también tener la posibilidad de convertir la información a otros formatos más flexibles y fácil de trasladar.



**Fig.12** Diagrama de despliegue que muestra todos los activos físicos que pueden o deben estar relacionados en el desarrollo de nuestra herramienta.

## 2.8 Vistas del reporte creado

Mostramos un croquis de la salida del reporte, no existe la posibilidad de cambiar el estándar, sería en ese orden, o sea, título, descripción (opcional), un primer bloque donde estarán las tablas con los datos de la información, luego los sumarios en caso de que algún usuario lo necesite, un segundo bloque, que aunque no se muestra tendría la misma estructura que el primero, y así sucesivamente pueden existir tantos bloques y tablas cree el cliente. Además, podemos agregar estilo de diseño al reporte, que sería incluido por medio de una CSS dentro del documento XML que definimos con anterioridad.

Válido aclarar que el color de las celdas del croquis se inserta para hacer agradable la tabla, a la vista del usuario, no significa que se mostrará el reporte con dicho diseño.

Título del reporte																	
Descripción																	
<table> <tr> <th colspan="3">Tabla 1</th></tr> <tr> <td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td></tr> <tr> <td>Sumario 1</td><td>Sumario2</td><td>Sumario3</td></tr> </table>			Tabla 1												Sumario 1	Sumario2	Sumario3
Tabla 1																	
Sumario 1	Sumario2	Sumario3															
Sumario general (tabla 1)																	
<table> <tr> <th colspan="3">Tabla 2</th></tr> <tr> <td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td></tr> <tr> <td>Sumario 1</td><td>Sumario2</td><td>Sumario3</td></tr> </table>			Tabla 2												Sumario 1	Sumario2	Sumario3
Tabla 2																	
Sumario 1	Sumario2	Sumario3															
Sumario general (tabla 2)																	
Sumario general del reporte																	

**Tabla 4:** Croquis del formato de salida del reporte.

Es importante haber entendido todos los aspectos que se han expuesto anteriormente, o al menos tener una idea de cómo funciona esta herramienta, porque además se incluyen las opciones de exportar el reporte a formato PDF y Excel. Para ello utilizamos dos librerías; una de ellas es *PHPExcel* que lee y escribe archivos con éste formato; y la librería *TCPDF*, que es una clase de php para crear documentos en formato PDF. Una excelente opción porque es software libre y la más usada en la actualidad por su eficiencia. Dentro de sus principales características y las que nos interesan se incluyen:

- Contiene 51 ejemplos listos para utilizar.
- No necesita de librerías externas o componentes para poder funcionar.
- Soporte para documentos encriptados.
- Soporte para impresión de código de barras.
- Incluye métodos para imprimir el contenido de una tabla.

Veamos a continuación una muestra de salida del reporte que proporciona CreativeReport.

### Reporte de películas

estadísticas de películas francesas

nombre pelicula	año producción	pais
Premiere seance	1995	Francia
Gazon maudit	1995	Francia
Misérables du vingtieme siècle, Les	1995	Francia
Fils de Gascogne, Le	1995	Francia
Mon homme	1995	Francia

[Ver reporte Completo](#) - [Exportar a Excel](#) - [Exportar a PDF](#)

**Fig.19:** Muestra la salida de un determinado reporte, en este caso solo muestra un resumen previo.

Como podemos ver existen tres links en la parte inferior del reporte, la opción *Ver reporte Completo* nos muestra la salida al cien por ciento del informe deseado.

## Reporte de películas

estadísticas de películas francesas

### Películas francesas

nombre película	año producción	pais
Premiere seance	1995	Francia
Gazon maudit	1995	Francia
Misérables du vingtieme siècle, Les	1995	Francia
Fils de Gascogne, Le	1995	Francia
Mon homme	1995	Francia
Marthe	1995	Francia
Citizen Langlois	1995	Francia
Histoires(s) du cinema	1995	Francia
Gérard Philippe, un prince dans le foule	1995	Francia
Alain Cavalier, 7 chapitres, 5 chambres et 2 pièces-cuisine	1995	Francia
Citizen Langlois	1995	Francia
Convoi, Le	1995	Francia
Hora de la piscina, La	1995	Francia
Haine, Le	1995	Francia
Shohei Imamura, le libre penseur	1995	Francia
Hussard sur le toit, Le	1995	Francia

[Ver reporte Completo](#) - [Exportar a Excel](#) - [Exportar a PDF](#)

**Fig.20:** Salida de un determinado reporte de películas de la base de datos.

## **2.9 Conclusiones**

En el capítulo observamos a través del análisis, diseño e implementación las funcionalidades del producto desarrollado, y se determina que éstas fueron las más adecuadas de acuerdo a los requisitos tanto funcionales como no funcionales impuestos, de manera tal que se implementaron con la mayor simpleza posible para disminuir el consumo de recursos en cuanto a hardware.

# Capítulo 3

## Validación de CreativeReport

Capítulo compuesto específicamente por la validación de las principales funcionalidades que se ejecutan en nuestro programa.

### Introducción

Todo sistema de software debe ser probado antes de su entrega o puesta en funcionamiento, a este proceso ó fase se le denomina *prueba* o *testeo* de la aplicación. Es una de las etapas, en la vida del software, más importante y que en ocasiones en las que menos recursos se le asignan, y esto es una mala práctica que conlleva en muchos casos al fracaso del producto o a la no satisfacción del cliente. No siempre las pruebas garantizan que el software esté perfecto en aspectos funcionales, pero si detectan numerosas fallas que el desarrollador no encuentra y que ahora pueden ser corregidas y vueltas a probar.

### 3.1 Casos de prueba. Características.

Son un conjunto de condiciones o variables bajo las cuáles se determina si el requisito de una aplicación es parcial o completamente satisfactorio.

Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio, con el propósito de comprobar que todos los requisitos de una aplicación son revisados.

El propósito es especificar una forma de probar el sistema, incluyendo las entradas con las que se ha de probar, los resultados esperados y las condiciones bajo las que ha de probarse. Normalmente, un caso de prueba se

deriva de un caso de uso en el modelo de casos de uso o de una realización de caso de uso en el Modelo de Diseño. Con estos casos de prueba se validan los requerimientos funcionales del sistema.

Cualquier producto de ingeniería puede ser probado de una de dos formas:

- Conociendo la función específica para la que fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa.
- Conociendo el funcionamiento del producto, se puede desarrollar pruebas que aseguren que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han comprobado en forma adecuada.

### **3.1.1 Rasgos generales del caso de prueba *caja negra***

Los métodos de prueba de la caja negra se centran en los requerimientos funcionales del software. O sea, permite derivar conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa, más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructura de datos o en acceso a base de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

A diferencia de la prueba de la caja blanca, que se lleva a cabo previamente en el proceso de prueba, caja negra tiende ser aplicada en posteriores fases de



prueba, ya que intencionadamente ignora la estructura de control y concentra su atención en el dominio de la información.

**Ventaja:**

Permite identificar claramente las entradas y salidas además de estudiar las relaciones que existen entre ellas. Maximiza la eficiencia de los sistemas sin tener que introducirnos en todos los procesos del software, exceptuando cuando se presentan anomalías en las relaciones de entrada y salida, entonces nos vemos obligados a estudiar ese subsistema de forma más precisa.

**3.2 Pruebas de unidad realizadas a las principales funcionalidades de CreativeReport.**

Optamos por aplicar caja negra, pues nos centramos en los resultados que tiene la salida del reporte, ya que el objetivo nuestro no es probar funcionalmente todos los métodos procedimentales del producto, como lo amerita *caja blanca*, sino testear las principales funcionalidades que cumpla con el requisito; además de que caja negra intenta encontrar errores en estructuras de datos o en accesos a las bases de datos externas que es una de las bases que nos interesa validar.

**3.2.1: Caso de prueba del método renderPreview ():**

Caso de prueba	Objetivo de la Prueba	Datos de Entrada	Salida Esperada	Salida Obtenida	Observación
1	Mostrar en la página web un previa de reporte.	XML que contiene en sus etiquetas los datos pertinentes (título, descripción, subtítulo, la consulta a la base de datos, y el estilo previamente definido.)	Tabla de 5 tuplas con la información de la consulta realizada en el archivo XML.	Tabla de 5 filas con la información de la consulta realizada en el archivo XML.	

**Tabla 5.** Validación del método **renderPreview()**.

En este caso de prueba, se comprobó la confiabilidad y la correcta implementación del método **renderPreview()**, que garantiza una previa del reporte, por tal razón la salida es solamente 5 tuplas, por estar definido así al usuario se le muestra un antecedente de la información que verá en el reporte completo.

### 3.2.2: Caso de prueba del método: **render()**:

Caso de prueba	Objetivo de la Prueba	Datos de Entrada	Salida Esperada	Salida Obtenida	Observación
1	Mostrar en la página web el reporte completo.	-	Mostrar en tablas todas las consultas del archivo XML, con todos sus datos.	Mostrar en tablas todas las consultas del archivo XML, con todos sus datos.	Al hacer clic sobre el hipervínculo <i>Ver reporte completo</i> .

**Tabla 6:** Caso de prueba del método: **render()**

En este caso de prueba se comprobó de igual manera que el anterior, la robustez de su desarrollo y la confiabilidad del mismo. Esta funcionalidad es la que garantiza mostrar el reporte, por tanto, es el principal método del software

en la cuál hubo que realizar varias pruebas para verificar su robustez en cualquier caso que pudiera existir en la variable de entrada.

### **3.3 Conclusiones**

Las pruebas de unidad confirmaron la existencia de algunos errores y validaron el buen comportamiento de las clases que no tenían problemas. A través de ellas se interceptaron pequeños errores en el código fuente que no habían sido tratados.

Esta fase de testeo es realmente significativa, sin las pruebas que verifiquen las funcionalidades más relevantes, nuestro software es vulnerable, y eso es algo que con el desarrollo de la informática prácticamente no tendría sentido realizar, por lo tanto, la fase de prueba garantiza que el sistema sea robusto y eficiente, además de satisfacer los requerimientos del cliente convertido en el principal acervo para un desarrollador.

## Conclusiones

Al valorar los generadores de reportes estudiados notamos que todos son semejantes y que ninguno llega a compensar las expectativas para su uso en la EICTV, de ahí que nuestra herramienta ha tenido un efecto grato por parte del cliente al atenuar las deficiencias que existen en la obtención de reportes con información filtrada. Aunque solamente está capacitado para los directivos, la ventaja de optimizar parte de trabajo manual es catalogada como un paso adelante en la informatización para toda la escuela, además de aprovechar las tecnologías con mayor eficiencia.

Ciertamente, **CreativeReport** ha creado una condición laboral mejorada en este aspecto de obtener información específica en la web tan indispensable para la toma de decisiones, también se convierte en un eslabón para los cimientos del sistema virtual la EICTV.

## Recomendaciones

- Implementar la interfaz visual de **CreativeReport**.
- Extender el uso de ésta herramienta a las empresas ó entidades que necesiten mostrar reportes en el sistema web de Información.

## Referencias Bibliográficas

- **Leyva, Ernesto.** Implementación del módulo de diseño de reportes para el SCADA Guardián del Alba. Ciudad de La Habana, Universidad de las Ciencias Informáticas, junio 2009.
- **CrystalReports.ORG.** Crystal Reports XI from Business Objects. Disponible en: [http://www.businessobjects.com/products/reporting/crystalreports/start\\_xi.asp](http://www.businessobjects.com/products/reporting/crystalreports/start_xi.asp) [En línea] 11 de enero de 2007.
- **Eustaquio, Rangel.** PHPReports Manual, noviembre 21,2006.
- **JasperReports.** Home. Disponible en <http://www.jasperreports.sourceforge.net>
- **Visual Paradigm para UML:** Disponible en: <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>
- **Apache.org.** [En línea] 2005: Disponible en: <http://www.facilnet.net/matriz/web2/apache.html>
- **Una introducción a apache:** Disponible en: [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/)
- **Prado, Guillermo** .Universidad de la Rioja, Página creada para la asignatura de Sistemas Informáticos.

## **Bibliografía**

**JAMES JACOBSON, I. B., GRADY Y RUMBAUGH.** El Proceso Unificado de Desarrollo de Software. La Habana, 2004. p.

**LARMAN, C.** UML y patrones. Introducción al análisis y diseño orientado a objetos. 1. 2004. p.

**PRESSMAN, R. S.** Ingeniería de Software. Un enfoque práctico. 5. La Habana, 2005. p.

**HAREM. 2007.** Lighttpd vs. Apache, PhpMagazine. [En línea] 2007.

[http://www.phpmagazine.net/2007/02/lighttpd\\_vs\\_apache.html](http://www.phpmagazine.net/2007/02/lighttpd_vs_apache.html).

## Glosario de Términos

- **Diseño de informe:** Apariencia final con la cual será generada el informe.
- **XML (Extensible Markup Language)** Metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos, por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.
- **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.
- **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.
- **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.
- **GPL:** Son las siglas de **G**eneral **P**ublic **L**icense, Licencia Pública General, definida por la Fundación para el Software Libre (FSF) para proteger los derechos de copia del software libre.



- **CSV** (en inglés *comma-separated values*): Son un tipo de documento sencillo para representar datos en forma de tabla.
- **DTD** (*definición de tipo de documento*) *DTD* (siglas en inglés de *document type definition*) es una descripción de estructura y sintaxis de un documento XML.
- **SQL** es un lenguaje formal declarativo, para manipular información en una base de datos. SQL es un acrónimo (*Structured Query Language*).
- **OEM** Fabricante de Equipos Originales (*OEM*) es un fabricante de computadoras. A menudo cuentan con acuerdos con Microsoft para distribuir software.
- **C++** es un lenguaje de programación diseñado a mediados de los años 1980.