



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 1**

**Trabajo de diploma para optar por el título de Ingeniero en
Ciencias Informáticas.**

**TÍTULO: Sistema para el control y seguimiento de incidencias
técnicas y de seguridad en la Dirección de Producción
Audiovisual.**

AUTORES: Yaima García Ruiz.
Jasiel Pavón Leyva.

TUTOR: Lic. Ulises Llorente Pérez.

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

Declaración de Autoría

Por este medio se declara que Yaima García Ruiz y Jasiel Pavón Leyva, son estudiantes de la Universidad de las Ciencias Informáticas, siendo los únicos autores de la presente investigación titulada: “Sistema para el control y seguimiento de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.” y autorizamos a hacer uso de la misma en su beneficio a la FAR, la UCI y cualquier otro centro educacional.

Para que así conste firmamos la presente a los 3 días del mes de junio del año 2010 .

Firma de los autores:

Yaima García Ruiz

Jasiel Pavón Leyva

Firma del Tutor:

Lic. Ulises Llorente Pérez.



“Cuando se es joven, se crea. Cuando se es inteligente, se produce. No se adapta, se innova: la medianía copia; la originalidad se atreve.”

José Martí



Agradecimientos

A:

*Mi familia, en especial a mis padres.
A todos mis compañeros que estuvieron ahí cada vez que los
necesitaba dándome su apoyo incondicional.
A mi tutor por guiarnos y ayudarnos durante toda la
investigación.
A la profesora Eyllin que también nos ayudó y nos atendió cada
vez que íbamos a verla para que nos revisara lo que teníamos
echo.
A mis compañeras de cuarto Dayana y María del Carmen; a sus
novios Yordán y Pedro por hacer mi vida en el apartamento más
divertida con sus maldades y sus aires de jefes de apartamentos
y al final las que mandábamos ahí, éramos las mujeres.*

Yaima.

A:

*Mi familia, a la profesora Eyllin, al profesor Jorge, a mi tutor
Ulises y a Francisco por ayudarme en realización de la
investigación, a mis amigos y a todos lo que siempre supieron
que triunfaría.*

Jasiel.



Dedicatoria

A nuestros padres, quienes con su apoyo incondicional y amor han sabido guiarnos para que podamos lograr nuestros sueños. A nuestros hermanos, a nuestros familiares y a nuestros amigos que nos apoyan y ayudan muchísimo para conseguir nuestros objetivos.

El aprovechamiento de la tecnología de la información y las comunicaciones (TIC), que se encuentran en rápida evolución, ha adquirido importancia creciente para el mejoramiento de la gestión de información de las organizaciones. Gracias a las nuevas tecnologías, se dispone de nuevos soportes que facilitan su registro, su almacenamiento, y su recuperación. La Dirección de Producción Audiovisual carece de una adecuada gestión de los procesos de incidencias técnicas y seguridad, ya que estos se realizan de forma manual, por lo que la entidad evaluó sus necesidades para construir un sistema que se encargue del procesamiento de dicha información. Por tales motivos, la propuesta consiste en la construcción del sistema para el control y seguimiento de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual, con el objetivo de lograr un proceso eficiente que garantizará la agilización de los procesos de gestión de incidencias, la obtención de una reducción de los gastos producto del consumo de material de oficina, así como la disminución de los riesgos por pérdida y deterioro de la información, aumentando el nivel de seguridad de la misma. Para llevar a cabo la creación del sistema, se tuvo como guía la metodología ágil XP*, además de la utilización de PHP* como lenguaje de programación del lado del servidor y a HTML*, CSS* y JavaScript del lado del cliente, para almacenar los datos registrados en la aplicación se empleó como gestor de base de datos a MySQL y se utilizó el framework symfony para la construcción del sistema. Como resultado se obtuvo un prototipo funcional del sistema que satisface las necesidades del cliente, además de la documentación pertinente obtenida a lo largo de la investigación.



Contenido

Introducción.....	1
Capítulo 1.....	6
1.1 <i>Introducción</i>	6
1.2 <i>Marco conceptual</i>	6
1.3 <i>¿Por qué una aplicación web y no una aplicación de escritorio?</i>	9
1.4 <i>Sistemas similares</i>	9
1.4.1 <i>Necesidad de hacer un sistema para la gestión de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual</i>	12
1.5 <i>Fundamentación de la tecnología a utilizar</i>	13
1.5.1 <i>Lenguajes del lado del servidor</i>	13
1.5.2 <i>Lenguajes del lado del cliente</i>	15
1.5.3 <i>Sistemas Gestores de Base de Datos</i>	16
1.5.4 <i>Proceso de desarrollo de software</i>	18
1.5.5 <i>Lenguajes de modelado</i>	19
1.5.6 <i>Herramienta CASE</i>	21
1.5.7 <i>Framework (Marco de trabajo)</i>	23
1.6 <i>Conclusiones</i>	25
Capítulo 2.....	26
2.1 <i>Introducción</i>	26
2.2 <i>Descripción de la solución propuesta</i>	26
2.3 <i>Especificación de requisitos</i>	27
2.4 <i>Personas relacionadas con el sistema</i>	30
2.5 <i>Fase de Planificación</i>	31
2.5.1 <i>Historias de Usuarios (HU)</i>	31
2.5.2 <i>Estimación de esfuerzo por HU</i>	35
2.5.3 <i>Plan de iteraciones</i>	35
2.5.4 <i>Plan de duración de iteraciones</i>	36
2.5.5 <i>Plan de entregas</i>	37
2.6 <i>Conclusiones</i>	37
Capítulo 3.....	38
3.1 <i>Introducción</i>	38
3.2 <i>Arquitectura del sistema</i>	38
3.3 <i>Fase de diseño</i>	39
3.3.1 <i>Patrones de diseño</i>	39
3.3.2 <i>Tarjetas CRC</i>	40
3.3.3 <i>Diseño de la Base de Datos</i>	43
3.4 <i>Fase de codificación</i>	43
3.4.1 <i>Tareas de la ingeniería</i>	43
3.4.2 <i>Interfaz de Usuario</i>	50
3.4.3 <i>Diagrama de despliegue</i>	50
3.5 <i>Fase de pruebas</i>	50
3.5.1 <i>Pruebas de Aceptación</i>	51
3.6 <i>Conclusiones</i>	51
Capítulo 4.....	52

Índice

4.1 Introducción	52
4.2 Características del Proyecto	52
4.2.1 Entradas externas.....	52
4.2.2 Salidas externas	53
4.2.3 Consultas Externas.....	53
4.2.4 Archivos Lógicos Internos	54
4.2.5 Archivos de Interfaz Externos	54
4.3 Estimación inicial.....	55
4.4 Cálculo del esfuerzo nominal	56
4.5 Cálculo del esfuerzo ajustado	57
4.6 Cálculo del tiempo de desarrollo, cantidad de hombres y costo.....	58
4.7 Beneficios tangibles e intangibles.....	59
4.7.1 Tangibles.....	59
4.7.2 Intangibles.....	59
4.8 Análisis de costo y beneficios	60
4.9 Conclusiones	60
Conclusiones	60
Recomendaciones	62
Referencias	63
Bibliografía.....	65
Glosario de términos	66

Índice de tablas

Tabla de Operalización de las variables	4
Tabla # 1.1 Necesidad de implementar un nuevo sistema	13
Tabla # 2.1 Personas relacionadas con el sistema.....	30
Tabla # 2.2 HU Autenticar.....	31
Tabla #2.3 HU Gestionar Usuario.....	32
Tabla #2.4 HU Mostrar permiso.....	32
Tabla # 2.5 HU Gestionar incidencias técnicas.	32
Tabla # 2.6 HU Gestionar incidencias seguridad.....	33
Tabla # 2.7 HU Gestionar soluciones incidencias técnicas.....	33
Tabla # 2.8 HU Gestionar soluciones incidencias seguridad.....	34
Tabla # 2.9 HU Generar listado reportes incidencias.	34
Tabla # 2.10 HU Buscar Incidencias técnicas.....	34
Tabla # 2.11 HU Buscar Incidencias seguridad	35
Tabla # 2.12 Estimación de esfuerzo por HU.....	35
Tabla # 2.13 Plan de duración de iteraciones.....	36
Tabla # 2.14 Plan de entregas	37
Tabla # 3.1 Tarjeta CRC: Clase IdapEmpet.....	40
Tabla # 3.2 Tarjeta CRC: Clase sf_guard_user	41
Tabla # 3.3 Tarjeta CRC: Clase sf_guard_permission	41
Tabla # 3.4 Tarjeta CRC: Clase crudIncidenciaT.....	41
Tabla # 3.5 Tarjeta CRC Clase: crudIncidencias.....	41
Tabla # 3.6 Tarjeta CRC Clase: crudSolucionIt	42
Tabla # 3.7 Tarjeta CRC Clase: crudSolucionIs	42
Tabla # 3.8 Tarjeta CRC Clase: reportes incidencias.....	42
Tabla # 3.9 Tarjeta Clase: buscar_incidenciaT	43
Tabla # 3.10 Tarjeta Clase: buscar_incidenciaS	43
Tabla # 3.11 Historias de usuarios desarrolladas en la 1ra iteración.....	44
Tabla # 3.12 Tarea para crear la interfaz del sistema.....	44
Tabla # 3.13 Tarea: Mostrar formulario de autenticación	44

Tabla # 3.14 Tarea: Validar campos para autenticarse	45
Tabla # 3.15 Tarea: Diseño del formulario para gestionar usuarios	45
Tabla # 3.16 Tarea: Validar campos para gestionar usuarios	45
Tabla # 3.17 Tarea: Diseño del formulario para mostrar permisos.....	46
Tabla # 3.18 Tarea: Diseño del formulario para gestionar incidencias técnicas	46
Tabla # 3.19 Tarea: Validar campos del formulario para gestionar incidencias técnicas	46
Tabla # 3.20 Tarea: Diseñar el formulario para gestionar incidencias de seguridad	47
Tabla # 3.21 Tarea: Validar campos del formulario para gestionar incidencias seguridad.....	47
Tabla # 3.22 Tarea: Buscar incidencias técnicas.....	47
Tabla # 3.23 Tarea: Buscar incidencias seguridad	47
Tabla # 3.24 Historias de usuarios desarrolladas en la 2da iteración	48
Tabla # 3.25 Tarea: Diseñar el formulario para gestionar solución incidencias técnicas	48
Tabla # 3.26 Tarea: Validar campos del formulario para gestionar solución a incidencias técnicas	48
Tabla # 3.27 Tarea: Diseñar el formulario para gestionar solución a incidencias de seguridad.....	49
Tabla # 3.28 Tarea: Validar campos del formulario para gestionar solución a incidencias de seguridad	49
Tabla # 3.29 Historias de usuarios desarrolladas en la 3ra iteración.....	49
Tabla # 3.30 Tarea: Generar listado de incidencias	49
Tabla # 4.1 Entradas Externas	53
Tabla # 4.2 Salidas Externas	53
Tabla # 4.3 Consultas Externas.....	54
Tabla # 4.4 Archivos Lógicos Internos	54
Tabla # 4.5 Archivos de Interfaz Externos	54
Tabla # 4.6 Peso del Factor de Complejidad.....	55
Tabla # 4.7 Puntos de Función desajustados	55
Tabla # 4.8 Factor Escalar (Wi)	57
Tabla # 4.9 Multiplicadores de esfuerzo (EM)	58
Tabla # 4.10 Resultados alcanzados.....	59

Índice de figuras

Figura #1: Vocabulario de UML.....	20
Figura #2 Arquitectura Cliente – Servidor	39
Figura # 3 Plantilla global en Symfony	40
Figura # 4 Diagrama de Despliegue.....	50



Introducción

Las tecnologías de la información y las comunicaciones han logrado tan alto grado de desarrollo, que hoy están en la mayoría de las actividades del ser humano: en la educación, la industria, el comercio, las finanzas, la investigación, entre otros. Su uso es un factor determinante en los niveles de eficiencia y competitividad tanto a nivel empresarial como personal.

Cuba está consciente de la revolución tecnológica que acontece en el mundo y en este sentido ha identificado la necesidad de introducir en la sociedad toda esta Tecnología de la Información y las Comunicaciones (TIC), para lograr una cultura digital, lo que facilitaría a la sociedad cubana acercarse más al objetivo de un desarrollo sostenible.

El 12 de enero del 2000 es creado el Ministerio de la Informática y las Comunicaciones por el Comité Ejecutivo del Consejo de Ministros de la República de Cuba. Esta organización, se encarga de impulsar el proceso de informatización para toda la sociedad cubana. Su misión es impulsar, facilitar y ordenar el uso masivo de servicios y productos de las tecnologías de la información, las comunicaciones, la electrónica y la automatización para satisfacer las expectativas de todas las esferas de la sociedad. (1)

Disímiles son los ejemplos que se pueden citar que demuestran la materialización de una idea fundamentada en el desarrollo informático del país, entre ellos se engloba la creación de los Joven Club de Computación y Electrónica, que han permitido el acceso gratuito a las TIC a personas de todas las edades, principalmente jóvenes y niños. Se destaca sin duda alguna la creación de la Universidad de las Ciencias Informáticas (UCI), creada al calor de la Batalla de Ideas como un proyecto de la Revolución, con el único objetivo de informatizar todos los sectores de la nación cubana y desarrollar la industria del *software* para contribuir al desarrollo económico del país.

En el 2003 el Comandante en Jefe propone la creación del “Programa Luz Producciones¹”. El objetivo de este programa era la producción de multicopias de vídeos en formato VHS², destinadas a facilitar el trabajo en la alfabetización y universalización en la hermana República Bolivariana de Venezuela, y otros programas nacionales e internacionales como las misiones: Robinsón, Ribas y Sucre. Ya en el año 2004 se materializa el proyecto y comienza la construcción del Complejo Industrial de Multicopiado “Luz Producciones”, ubicado en el municipio de Marianao en Ciudad de La Habana. Surgió así una nueva industria en el país, con tecnología para el multicopiado en soporte VHS, CD y DVD.

¹ Programa Luz Producciones: su misión es duplicar los videos clases y otros materiales en soporte VHS, DVD y CD para la alfabetización, la universalización y otros programas educativos nacionales e internacionales.

² VHS: Soporte de video.

El Consejo de Estado, ponderando que la UCI cumple un papel importantísimo en el desarrollo de la Industria Cubana del *Software*, la materialización de los proyectos asociados a la informatización de la sociedad y por contar con el personal capacitado; toma la decisión de ubicar dentro de la UCI la realización de autorías de DVD para esta industria. Creándose así, el 1ro de octubre del 2006 la Dirección Casa Autoría DVD, subordinada a la Dirección General de la Infraestructura Productiva; actualmente cambió su nombre a Dirección de Producción Audiovisual y está subordinada al Centro de Tecnología para la Formación (FORTES) de la facultad 8.

La Dirección de Producción Audiovisual brinda servicios tales como:

- Autoría de DVD.
- Edición AVID en definición estándar y alta definición.
- Musicalización y realización de sonidos para audiovisuales.
- Conversión, digitalización y *transfer*³ de vídeo.
- Asesoría y producción de audiovisuales.

Por lo sensible de la información que se procesa en la entidad, se hace necesario la salvaguarda de esta, por tal razón se tiene instalado un sistema de Circuito Cerrado de Televisión y un Sistema Automatizado de Control de Acceso que controla y monitorea el acceso del personal a los departamentos.

Cuando ocurre algún incidente técnico o de seguridad, este es reportado a las personas encargadas. Estos procesos se realizan de forma manual, lo que provoca el deterioro de los documentos producto de su uso frecuente por ser el papel un material biodegradable, pérdida de la información, duplicación o modificación no autorizada de la misma, el cúmulo de documentos da paso a que se pierda tiempo buscando algún documento y a que estos ocupen gran cantidad de espacio físico, además del consumo de recursos, entre otros. A partir del análisis realizado se plantea el **problema científico**:

¿Cómo mejorar la gestión de información en los procesos de control y seguimiento de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual?

Este problema tiene como **objeto de estudio**: Los procesos de control y seguimiento de las incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.

El **campo de acción** abarcado es: La gestión de la información en los procesos de control y seguimiento de las incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.

³ *Transfer*: traspaso

Para darle solución al problema expuesto se define como **objetivo**:

Proponer una solución informática que permita la gestión de información en los procesos de control y seguimiento de las incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.

Para darle cumplimiento al objetivo propuesto, se plantean las siguientes **tareas de la investigación**:

- Realizar el estudio del estado del arte de los sistemas de gestión de incidencias.
- Definir los requerimientos y la planificación de la solución informática propuesta.
- Obtener el diseño y desarrollo de la solución informática propuesta.
- Realizar pruebas de aceptación al sistema.
- Realizar el estudio de la factibilidad para el desarrollo de la solución informática propuesta.

Se plantea como **hipótesis**: Si se informatizan los procesos de control y seguimiento de las incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual, se mejorará la gestión de información de los mismos.

Variable independiente: informatizar los procesos de control y seguimiento de incidencias técnicas y de seguridad.

Variable dependiente: la gestión de información en los procesos de control y seguimiento de las incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.

Variables	Dimensiones	Indicadores	Unidad de Medida
Sistema	Factibilidad	Tiempo de desarrollo	Extenso
			Moderado
			Breve
		Costo	Costoso
			Moderado
			Barato
		Esfuerzo	Alto
			Moderado
			Despreciable
		Rapidez	Alta
			Media

Proceso de gestión de información para el control de las incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual	Eficiencia	Seguridad	Baja
			Alta
			Media
			Baja
		Organización del trabajo	Buena
			Regular
	Cliente	Satisfacción con el sistema	Mala
			Alta
			Media
		Agilidad en el proceso	Baja
Rápido			
Medio			
			Lento

Tabla de Operalización de las variables

Para la realización de la investigación, se tuvo en consideración el siguiente sistema de métodos:

➤ **Métodos Teóricos:**

- ✓ Analítico – sintético: se utilizó para analizar la información relacionada con el objeto de estudio obtenido en la investigación.
- ✓ Análisis histórico - lógico: permitió conocer sobre el estado del arte del objeto de estudio, analizar su evolución y desarrollo.

➤ **Métodos Empíricos:**

- ✓ Observación: permitió una visión de la realidad durante el desarrollo de la investigación.
- ✓ Entrevista: permitió obtener la información necesaria relacionada con el proceso de reportes de las incidencias técnicas y de seguridad, mediante entrevistas realizadas a personas encargadas del tema.
- ✓ Estudio documental: sintetizó lo referente al objeto de estudio de la investigación.

Estructura de la investigación

El documento cuenta con 4 capítulos:

- **Capítulo 1. Fundamentación Teórica:** fundamenta el por qué hacer un nuevo sistema para la gestión de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual y cuáles son las herramientas, lenguajes, metodologías, frameworks y Sistemas Gestores de Base de Datos que se utilizan para el desarrollo del mismo.
- **Capítulo 2: Descripción de la solución propuesta:** describe la solución propuesta para resolver el problema planteado en la investigación, especificando los requerimientos funcionales y no funcionales y definiendo las historias de usuarios del sistema. Además, se realiza la estimación de esfuerzo asociada a la implementación de las historias de usuarios, estableciendo así, un plan de entrega para cada una de las iteraciones.
- **Capítulo 3: Construcción de la solución propuesta:** se definen las técnicas de modelado que ayudarán a los desarrolladores del *software*, a crear diseños de clases orientados a responsabilidades (tarjetas CRC). Como parte del plan se descomponen las historias en tareas de la ingeniería. Además, por cada historia de usuario, se realizan las pruebas de aceptación para verificar si al final de cada iteración se obtuvo la funcionalidad requerida.
- **Capítulo 4: Estudio de la factibilidad:** contiene el estudio de factibilidad realizado al sistema, enumerando los beneficios tangibles e intangibles y analizando los costos y beneficios que representa la elaboración de la aplicación.

Aporte Práctico: Una solución informática que permite la gestión de información en los procesos de control y seguimiento de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.

Nota: El huso del operador * seguido de una palabra es para identificar que la palabra se encuentra en el glosario de términos.

Capítulo 1: *Fundamentación teórica*

Capítulo 1

1.1 Introducción

En este capítulo se plantea la necesidad de implementar un sistema para mejorar la gestión de las incidencias técnicas y de seguridad que ocurren en la Dirección de Producción Audiovisual. Quedan definidos conceptos relacionados con el problema en cuestión; se realiza un estudio sobre los sistemas existentes que gestionan información sobre incidencias técnicas y de seguridad y se justifica la selección de las herramientas, tecnologías, metodologías y lenguajes con las que se elabora la investigación.

1.2 Marco conceptual

Para realizar una buena gestión de información en una organización, se debe tener claro, entre otras cosas, en que nivel de procesos administrativos se trabaja y que se espera de la investigación. Sin embargo, antes de hacer cualquiera de estas preguntas, se debe estar seguro de los conceptos primarios del ámbito de acción y al mismo tiempo, que la organización en la que se está prestando el servicio, concibe del mismo modo estos conceptos.

Análisis y Conceptualización:

- Según el gran diccionario de la lengua española en formato digital, **gestión** es, la organización, utilización y aprovechamiento de determinados dispositivos y recursos de un ordenador.
- **Gestión:** Actividades coordinadas para dirigir y controlar una organización. (2)

Teniendo en cuenta los conceptos anteriores, se puede decir que la **gestión** se refiere al desarrollo de las funciones básicas de la administración: planear, organizar, dirigir y controlar.

¿A qué nivel se va a gestionar?

¿A nivel de datos o de información?

Antes de responder estas interrogantes se debe tener claro que son los datos, la información, la gestión de datos y la gestión de información.

- **Datos:** elemento primario de la información conformada por símbolos (letras, números, dibujos, señas, gestos) que reunida puede cobrar significación. Solo o aislado el dato no posee relevancia, pero utilizado en las premisas de un razonamiento puede llevarnos a una conclusión. (3)

Capítulo 1: *Fundamentación teórica*

- Para la informática, los **datos** son expresiones generales que describen características de las entidades sobre las que operan los algoritmos. Estas expresiones deben presentarse de una cierta manera, para que puedan ser tratadas por una computadora. En este caso, los datos por sí solos tampoco constituyen información, sino que ésta surge del adecuado procesamiento de los datos. (4).
- Técnicamente, los **datos** son hechos y cifras en bruto, tales como órdenes y pagos, los cuales se procesan para obtener información, por ejemplo el saldo deudor y el monto disponible. (5)

Teniendo en cuenta los planteamientos anteriores, se define como **datos** a una expresión mínima de contenido respecto a un tema. Un conjunto de datos constituye una información.

- Para Ferrell y Hirt, la **información** "comprende los datos y conocimientos que se usan en la toma de decisiones"
- Según Idalberto Chiavenato, **información** "es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones" (6)

Teniendo en cuenta las anteriores ideas y definiciones, se plantea que la **información** es un conjunto de datos acerca de algún suceso, hecho o fenómeno, que organizados en un contexto determinado tienen un significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo.

- **Gestión de la información:** proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes, y que gestiona el acceso y los derechos de los usuarios sobre la misma. (7)
- **Gestión de información:** Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada. (8)

Capítulo 1: *Fundamentación teórica*

Teniendo en cuenta los criterios anteriores se plantea que la **gestión de información**, es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de la organización.

- Una de las más grandes organizaciones que se ocupan de la gestión de datos, DAMA (Asociación de Gestión de Datos), establece que la **gestión de los datos** es el proceso de desarrollo de arquitecturas de datos, prácticas y procedimientos que gestionan apropiadamente las necesidades del ciclo de vida completo de los datos de una empresa. (9)

Después de haber analizado cada uno de los términos de gestión de datos y gestión de información, se decide gestionar a nivel de información.

- **Incidencia:** es cualquier evento que no forma parte de la operación estándar de un servicio y que causa, o puede causar una interrupción, o una reducción de la calidad del mismo.
- **Incidencia técnica:** es la ocurrencia de mal funcionamiento o rotura de algún equipo tecnológico, sus accesorios y utilitarios.
- **Incidencia de seguridad:** cualquier evento que se produzca, de forma accidental o intencional, que afecte o ponga en peligro las tecnologías de información o los procesos que con ellas se realizan. (10)
- **La gestión por procesos** es la forma de gestionar toda la organización basándose en los procesos. Entiéndase por **proceso:** conjunto de recursos y actividades interrelacionadas que transforman elementos de entrada en elementos de salida. Los recursos pueden incluir personal, finanzas, instalaciones, equipos, técnicas y métodos. (11)
- En la Dirección de Producción Audiovisual, se realizan **procesos de control y seguimiento de incidencias técnicas y de seguridad**, estos no son más que un conjunto de actividades o eventos con el propósito de llevar el control y de darle seguimiento a las incidencias técnicas y de seguridad que ocurran en la entidad.
- Partiendo de los conceptos particulares a los más generales, se define como **proceso de gestión de información para el control y seguimiento de incidencias técnicas y de seguridad**, al conjunto de actividades relacionadas entre sí, a las que se les da una

Capítulo 1: *Fundamentación teórica*

manipulación y un tratamiento adecuado que permite el control de las incidencias técnicas y de seguridad que ocurren en la Dirección de Producción Audiovisual.

Actualmente estos procesos se realizan de forma manual, es por ello, que con el fin de mejorar la gestión de información de dichos procesos, se propone realizar una solución informática. Dicha solución sería el desarrollo de una aplicación web.

1.3 ¿Por qué una aplicación web y no una aplicación de escritorio?

Se decide realizar una aplicación web y no una de escritorio, porque en las aplicaciones de escritorios el despliegue y mantenimiento resulta una tarea bastante compleja y engorrosa; para su instalación es necesario ir a cada usuario que utilizará dicho *software* e instalar éste en cada máquina cliente, esto trae consigo numerosos inconvenientes, por ejemplo, una mejora en el servidor, como parte de la aplicación, requiere normalmente una mejora de los clientes instalados en cada ordenador personal, añadiendo un coste de soporte técnico y disminuyendo la productividad.

En las aplicaciones web, por el contrario, tanto el *software* como los datos se alojan en una máquina diferente (remota) y centralizada, para trabajar de manera estable, continua e ininterrumpida. El usuario se conecta a este servidor a través de un simple navegador web dotado de interactividad para la gestión, una base de datos, comunicaciones encriptadas y contraseñas de acceso.

En este tipo de aplicación el gasto se reduce, puesto que su mantenimiento se concentra en el servidor. Es su proveedor quien se preocupa por mantenerla siempre disponible y actualizada a cambio de una cuota fija y razonable.

Ventajas:

- Permite contar con clientes independientes de tecnología y estaciones de trabajo
- Permite una administración de forma centralizada.
- Permite tener la información centralizada eliminando la necesidad de hacer replicas periódicas.
- Una reducción del capital humano para el mantenimiento del sistema.
- Brinda la posibilidad de actualizar y mantener las aplicaciones sin la necesidad de distribuir e instalar software en un variado número de clientes.
- Permite la navegación desde cualquier punto.
- Permite ahorrar al no tener que invertir en costosas infraestructuras de comunicación que en muchos casos podrían ser sencillamente imposibles.
- Posibilita que el cliente cuente con las actualizaciones en el instante en que se realizan.
- Necesita solamente de un navegador Web para acceder al sistema.

Capítulo 1: *Fundamentación teórica*

1.4 Sistemas similares

➤ XPERTA

Xperta es una aplicación desarrollada por ASAC Comunicaciones, con el propósito de apoyar a los servicios técnicos en la gestión de sus incidencias. La aplicación es utilizada en el centro de atención a usuarios de la Universidad de Oviedo de España, como herramienta de gestión de servicios. En dicha entidad se utiliza la gestión de incidencias entre varios proveedores y entre el propio personal de la Universidad, siendo la herramienta utilizada para el control y seguimiento de los servicios prestados por proveedores externos a la Universidad de Oviedo.

Ventajas:

- ✓ Resuelve de forma rápida el proceso que supone la notificación de una incidencia por parte de un usuario o cliente y su comunicación posterior al equipo técnico correspondiente.
- ✓ Ofrece la posibilidad de que sus clientes registren directamente las incidencias en el sistema a través de una interfaz web, de esta forma, la incidencia queda registrada para luego ser tratada por los técnicos responsables del servicio.
- ✓ Los técnicos de un vistazo pueden ver el estado del servicio y los avisos o incidencias que tienen pendientes, disponiendo de información instantánea sobre el usuario que comunica la incidencia, sus datos y el historial de las incidencias comunicadas anteriormente.
- ✓ Ofrece información detallada acerca del estado del servicio, balance de los trabajos realizados, tipos de problemas más comunicados, etc.
- ✓ Obtiene un seguimiento total de las incidencias, desde el momento en que el usuario la comunica hasta llegar a su solución. (12)

Desventajas:

- ✓ No gestiona incidencias de seguridad.
- ✓ Los datos que gestionan no son los mismos que los que se gestionarán en la Dirección de Producción Audiovisual.

➤ Sistema de Gestión de Incidencias. Inventario de aplicativos de las CC. AA

El Sistema controla y mantiene informado sobre el ciclo de vida de las incidencias. Este sistema se usa para gestionar los diferentes problemas y las posibles mejoras de las plataformas propietarias de la Junta de Andalucía en España, que forman parte de la infraestructura de la administración electrónica. La utilización de este sistema permite además de agilizar la operativa de gestión de incidencias, obtener informes de operación, a partir de los cuales pueden medirse parámetros de calidad en la resolución de dichas incidencias. (13) A pesar de gestionar incidencias técnicas, no

Capítulo 1: *Fundamentación teórica*

gestiona incidencias de seguridad y no cumple con los requisitos que se necesitan en la entidad para la gestión de incidencias técnicas y de seguridad.

➤ **Service Desk**

Service Desk es una aplicación que facilita la gestión de incidencias desde su registro inicial hasta su cierre, incorpora estándares internacionales de buenas prácticas como ITIL (Information Technology Infrastructure Library)⁴. La herramienta permite minimizar los períodos de fuera de servicio, registrar la información relevante de todas las incidencias e incorporar las mejores prácticas del mercado de forma sistemática. (14) Esta herramienta no cumple con los requisitos necesitados por la entidad, además no gestiona incidencias de seguridad.

➤ **Equipo de Respuesta a Incidentes Computacionales de Cuba (CuCERT)**

La Oficina de Seguridad para las Redes Informáticas (OSRI) es una entidad nacional adscrita al Ministerio de la Informática y las Comunicaciones, que tiene por objeto social llevar a cabo la prevención, evaluación, aviso, investigación y respuesta a las acciones tanto internas como externas que afecten el normal funcionamiento de las Tecnologías de la Información del país. Cuenta con un sistema (CuCERT) mediante el cual se pueden reportar incidencias de seguridad informática desde cualquier lugar de Cuba. CuCERT tiene como misión prevenir y responder a los incidentes computacionales en el país. Entre sus principales objetivos, está informar sobre vulnerabilidades y amenazas de seguridad informática que se presenten en el país así como detectar y dar respuesta a dichos incidentes computacionales. Este sistema no gestiona incidencias técnicas, por este motivo no puede ser utilizado como solución para la gestión de incidencias en la Dirección de Producción Audiovisual.

➤ **UCICert**

Es un sistema mediante el cual se reportan incidentes computacionales en la Universidad de las Ciencias Informáticas (UCI). Cuenta con un equipo de respuestas ante tales incidentes que está integrado por un grupo de profesionales pertenecientes a la Dirección de Redes y Seguridad Informática. Tiene como misión prevenir y responder a los incidentes computacionales en la

⁴ La metodología ITIL (Information Technology Infrastructure Library) es una colección de las mejores prácticas observadas en el sector de TI que se ha convertido en un estándar “de facto”. ITIL describe los procesos de Gestión de Servicios de TI y se puede adaptar a cualquier departamento de informática. Su objetivo es restablecer el funcionamiento normal del servicio lo más rápidamente posible, y con el menor impacto sobre la actividad del negocio.

Capítulo 1: *Fundamentación teórica*

universidad y entre sus principales objetivos está detectar cualquier incidente de seguridad informática, informar sobre vulnerabilidades y amenazas y responder ante los incidentes que se presenten en la Universidad. Este sistema no gestiona incidencias técnicas, además los datos que maneja no son los mismos que se gestionarán en la Dirección de Producción Audiovisual.

➤ **GatServer**

Es un sistema que tiene entre sus principales funcionalidades, gestionar todos los datos referentes a las incidencias técnicas que ocurren en la Universidad de las Ciencias Informáticas. Su utilización permite almacenar de manera clara, sencilla y ordenada los datos referentes a las incidencias, así como una mayor velocidad en el proceso de reporte. Además, la aplicación cuenta con un equipo capacitado para darle solución a dichos incidentes. A pesar de todas estas ventajas, el sistema no puede ser usado como una solución para gestionar incidencias en la Dirección de Producción Audiovisual, porque no gestiona incidencias de seguridad y los datos que gestiona no son compatibles con los de la entidad.

1.4.1 Necesidad de hacer un sistema para la gestión de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual

Después de haber realizado un estudio de algunos sistemas existentes en el mundo, en Cuba y en la universidad que gestionan información relacionada con las incidencias técnicas y de seguridad, se decide implementar un nuevo sistema, pues ninguno de ellos, de forma general, cumple con los requerimientos que se necesitan para gestionar incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.

Capítulo 1: *Fundamentación teórica*

Tabla resumen

Parámetros que debe cumplir el sistema a realizar	Sistemas similares					
	XPERTA	Sistema de Gestión de Incidencias. Inventario de aplicativos de las CC. AA	Service Desk	Equipo de Respuesta a Incidentes Computacionales de Cuba (CuCERT)	UCICert	GatServer
Aplicación web	X	X	X	X	X	X
Gestionar incidencias técnicas	X	X	X	-	-	X
Gestionar incidencias de seguridad	-	-	-	X	X	-
Compatibilidad de datos	-	-	-	-	-	-

Tabla # 1.1 Necesidad de implementar un nuevo sistema

1.5 Fundamentación de la tecnología a utilizar

1.5.1 Lenguajes del lado del servidor

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el servidor y que se envían al cliente en un formato comprensible para él. Estos lenguajes son independientes del cliente, por lo que son mucho menos rígidos respecto al cambio de un navegador a otro o respecto a las versiones del mismo. Algunos de los más usados para el desarrollo de páginas dinámicas son: ASP*, PHP* y PERL*. (15)

➤ ASP*

ASP* (“*Active Server Pages*” o *Páginas Activas de Servidor*) es una tecnología desarrollada para la creación de páginas dinámicas e interactivas. Las páginas ASP* tienen por objetivo, generar *script* del lado del servidor, ejecutándose estos antes de ser enviados. Esta tecnología permite crear

Capítulo 1: *Fundamentación teórica*

aplicaciones independientes de la plataforma que se utiliza en el navegador.

Una de las características más importantes es que esta tecnología pertenece a la parte servidor, por esto no es necesario que el cliente o navegador la soporte.

➤ **PHP***

PHP* (*"Hypertext Preprocessor"*) es un lenguaje de programación del lado del servidor, es rápido, simple de usar, tiene una gran librería de funciones y mucha documentación, además es muy simple acceder a bases de datos desde él.

Puede ser desplegado en la mayoría de los servidores Web, en casi todos los sistemas operativos y plataformas sin coste alguno, esto último se debe a que es un lenguaje de código abierto, o sea, no se necesita pagar licencia. Entre las principales ventajas se pueden señalar:

- ✓ Es un lenguaje que puede ejecutarse en diferentes sistemas operativos (Mac OS, Unix/Linux, Windows).
- ✓ Permite programar sitios con contenido dinámico, combinándolo con los principales servidores de bases de datos como son: MySQL, PostgreSQL, Oracle y otros.
- ✓ Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ✓ No requiere definición de tipos de variables.
- ✓ Tiene manejo de excepciones (desde PHP* 5).
- ✓ Es un lenguaje libre y gratuito, está amparado bajo el movimiento open source (código abierto).

➤ **PERL***

PERL*⁵ (*"Practical Extraction and Report Language"* o *lenguaje práctico de extracción y de informes*) es un lenguaje interpretado que compila los programas antes de ejecutarlos, razón por la cual se habla de *scripts*⁵ y no de programas⁶.

Es un lenguaje multiplataforma, esto se debe a que los *scripts* son compatibles con disímiles plataformas.

⁵ *Scripts*: son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución, se distinguen de los programas, pues deben ser convertidos a un archivo binario ejecutable para correrlos.

⁶ Programas: referido principalmente a programas compilados al lenguaje máquina nativo del ordenador y sistema operativo en el que se ejecuta.

Capítulo 1: *Fundamentación teórica*

Entre sus principales características se pueden señalar que es fácil de usar, soporta tanto programación estructurada como programación orientada a objetos y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de textos y una enorme colección de módulos disponibles.

PERL* se puede usar para diferentes cosas como tratamientos y generación de ficheros de texto, simplificar las tareas de administración de UNIX, acceder a bases de datos gratuitas mediante módulos adicionales tales como el DBD (Diseñador de Bases de Datos) o el ODBC (*Open DataBase Connectivity*), esto último se puede combinar con un CGI (*Common Gateway Interface*) para hacer aplicaciones tales como un carrito de la compra para un sitio web, etc. (16)

➤ JSP

JSP (“Java Server Pages”) es una tecnología orientada a crear páginas web con programación en Java. Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Por tanto, las JSP se pueden escribir con el editor HTML*/XML habitual. (15)

Selección del lenguaje del servidor

Se escogió como lenguaje de programación del lado del servidor PHP* por su simpleza, por ser libre y gratuito, lo que permite sacar provecho de sus beneficios de manera gratuita, razón por la cual es uno de los lenguajes de programación más utilizados en la actualidad.

1.5.2 Lenguajes del lado del cliente

Los lenguajes del lado del cliente son aquellos que pueden ser directamente “digeridos” por el navegador y no necesitan un pre-tratamiento.

➤ HTML*

HTML* (“*HyperText Markup Language*”) es el lenguaje de marcado predominante para la construcción de páginas web, indica al navegador donde colocar cada texto, cada imagen o cada vídeo y la forma que tendrán estos al ser colocados en la página.

Cuenta de etiquetas rodeadas por corchetes angulares (<,>) y cada una significa una cosa, por ejemplo: significa que se escriba en negrita (bold) o <P> significa un párrafo, etc.; casi todas las etiquetas tienen su correspondiente etiqueta de cierre (/>), que indica que a partir de ese punto no debe de afectar la etiqueta, ejemplo se utiliza para indicar que se deje de escribir en negrita. Así que el HTML* no es más que una serie de etiquetas que se utilizan para definir la forma o estilo que se quiere aplicar al documento. (15)

Capítulo 1: *Fundamentación teórica*

➤ JavaScript

JavaScript es un lenguaje de programación del lado del cliente, porque el navegador es el que soporta la carga de procesamiento.

Se utiliza para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web.

Al contrario de Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, al no disponer de herencia; es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. (17)

➤ CSS*

CSS* (*"Cascading Style Sheets" u Hoja de Estilo en Cascada*) es un lenguaje de hojas de estilos que controla el aspecto o presentación de los documentos electrónicos definidos con HTML* y XHTML siendo la mejor forma para separar los contenidos y su presentación. Al separar la definición de los contenidos y la definición de su aspecto se mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Una vez creados los contenidos, se utiliza el lenguaje CSS* para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página.

Selección del lenguaje del lado del cliente

Se decidió utilizar los tres lenguajes (HTML*, CSS* y JavaScript). Los lenguajes HTML* y CSS* permitirán manejar con más facilidad todo lo referente al diseño, ya que HTML* indica al navegador donde colocar cada texto e imagen y la forma que tendrán estos al ser colocados en la página; con CSS* se podrá incluir márgenes, tipos de letra, fondos, colores, incluso definir los estilos en un archivo externo a las páginas de la interfaz, logrando así la actualización automática de todas las páginas del sitio web, en caso que se desee cambiar algún estilo; y con JavaScript se podrán hacer las validaciones de los campos de entradas de datos.

1.5.3 Sistemas Gestores de Base de Datos

Un SGBD* es un conjunto de programas que administran y gestionan la información contenida en una base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto, debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD*

Capítulo 1: *Fundamentación teórica*

- Manipular la base de datos: realizar consultas, actualizarla y generar informes.
- Conservar la integridad de los datos.
- Controlar la seguridad y privacidad de los datos. (18)

➤ **MySQL**

MySQL es un sistema de gestión de base de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del *software* libre, por su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son:

- ✓ Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multihilo.
- ✓ Soporta gran cantidad de tipos de datos para las columnas.
- ✓ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP*, etc.).
- ✓ Gran portabilidad entre sistemas.
- ✓ Soporta hasta 32 índices por tabla.
- ✓ Gestión de usuarios y *passwords*, manteniendo un muy buen nivel de seguridad en los datos.
- ✓ Velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
- ✓ Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- ✓ Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- ✓ Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor sino en el sistema en el que está.
- ✓ El conjunto de aplicaciones Apache-PHP*-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net). (19)

➤ **PostgreSQL**

PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Capítulo 1: *Fundamentación teórica*

Entre sus principales características se destacan:

- ✓ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
 - ✓ Incorpora una estructura de datos *array*.
 - ✓ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
 - ✓ Permite la declaración de funciones propias, así como la definición de disparadores.
 - ✓ Soporta el uso de índices, reglas y vistas.
 - ✓ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen).
-
- ✓ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos. (20)

Selección del sistema gestor de base de datos

Se decidió utilizar MySQL, porque además de ser uno de los gestores de bases de datos más usado en el mundo del *software* libre, es fácil de usar y muy rápido. Además, se instaló el WampServer, que es un contenedor en el que viene instalado MySQL, Apache y PHP*.

1.5.4 Proceso de desarrollo de *software*

Un proceso de desarrollo de *software* es un conjunto de actividades encaminadas a transformar los requisitos del cliente en un sistema de *software*.

➤ **XP***

La programación extrema (XP*) es una metodología de desarrollo de *software* que trata de dar al cliente lo que él quiere y cuando lo necesita; se basa en la comunicación, la simplicidad y el reciclado continuo de código; potencia al máximo el trabajo en equipo, debido a que tanto el jefe de proyecto, los desarrolladores y el cliente, están involucrados en el desarrollo del *software*. XP* es una metodología más liviana y ágil, está orientada más a las personas que a los procesos ya que supone que:

- ✓ Las personas son la clave en los procesos de desarrollo.
- ✓ Los programadores son profesionales, no necesitan supervisión.
- ✓ Los procesos se aceptan y se adecuan, no se imponen.
- ✓ Los desarrolladores y gerentes comparten el liderazgo del proyecto.
- ✓ El trabajo con las personas que conocen el negocio es regular, no puntual.

Capítulo 1: *Fundamentación teórica*

➤ RUP*

La metodología RUP* es un proceso de desarrollo de *software* que unifica los mejores elementos de metodologías anteriores, el cual está preparado para el desarrollo de proyectos complejos y grandes, y es orientado a objetos. Asume tres características fundamentales que lo diferencian del resto de las metodologías:

- ✓ **Dirigido por casos de uso:** Significa que el proceso sigue un hilo a través de flujos de trabajo que parten de los Casos de Uso. Guían el diseño, la implementación y las pruebas.
- ✓ **Centrado en la arquitectura:** Son las vistas en cada flujo de trabajo, que contiene los elementos fundamentales en cada uno de ellos.
- ✓ **Iterativo e incremental:** Es la estrategia para desarrollar un producto de *software* en pasos pequeños y manejables. Al iterar se decide si continuar o no con el proyecto, y se sabe si hubo incremento.

➤ Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas, para trabajar en equipo y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras, y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. Está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad y la productividad son fundamentales. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente, cuando este lo solicite. (21)

Selección de la metodología a utilizar

Se decidió utilizar XP* debido a que se adapta en gran medida, tanto al tipo del proyecto a desarrollar como a las condiciones de trabajo. La planificación es más transparente, pues los clientes forman parte del equipo de trabajo y no tiene sobrecarga de artefactos, que en ocasiones resulta molesta, además, esta metodología se adapta perfectamente a los proyectos cuyos requerimientos cambian a menudo.

1.5.5 Lenguajes de modelado

El lenguaje de modelado de objetos, es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar (parte de) un diseño de *software* orientado a objetos.

Capítulo 1: *Fundamentación teórica*

➤ UML*

Lenguaje Unificado de Modelado (*UML**, por sus siglas en inglés “*Unified Modeling Language*”) es el lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML* ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

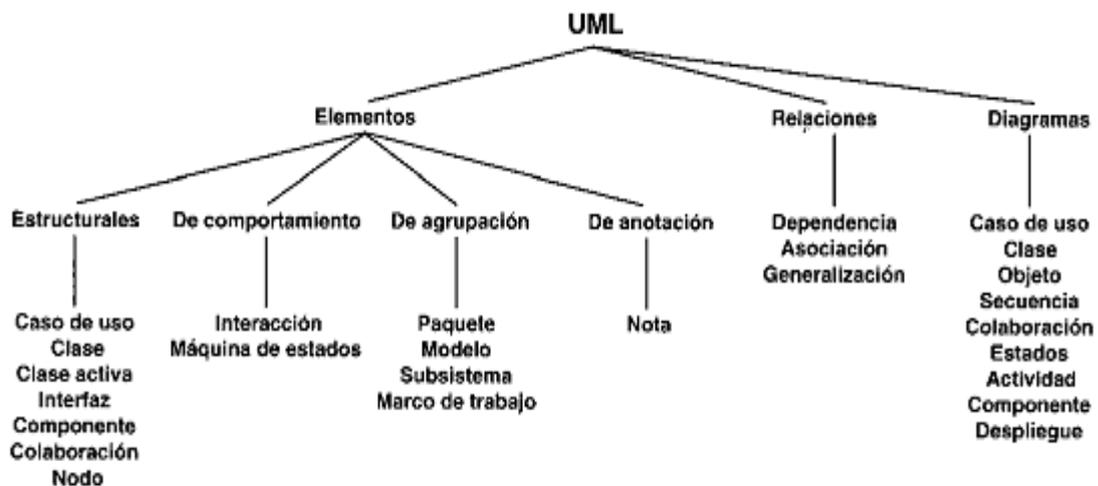


Figura #1: Vocabulario de UML*

UML* cuenta con varios tipos de diagramas que muestran diferentes aspectos de las entidades representadas, dichos diagramas son:

- ✓ Diagramas de Estructura: enfatizan en los elementos que deben existir en el sistema modelado.
- ✓ Diagramas de Comportamiento: enfatizan en lo que debe suceder en el sistema modelado.
- ✓ Diagramas de Interacción: son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado.

➤ Lenguaje de Modelado para Realidad Virtual (VRML)

VRML sirve para crear mundos en tres dimensiones a los que accedemos utilizando nuestro navegador, igual que si visitásemos una página web cualquiera, con la salvedad que nuestras visitas no se limitan a ver un simple texto y fotografías, sino que nos permite ver todo tipo de objetos y construcciones en 3D por los que podemos pasear o interactuar, tratar con los objetos como en la vida misma, tocarlos, arrastrarlos, etc. VRML permite modelar objetos tridimensionales y su comportamiento.

Capítulo 1: *Fundamentación teórica*

Este lenguaje de modelado describe escenas, donde cada una, está compuesta por objetos. Los objetos pueden ser formas sólidas o elementos intangibles que afectan la escena, ejemplo: luces, sonidos, etc. Un documento VRML está compuesto por nodos, estos van a ser la estructura mínima de un fichero y su misión es definir las características de un objeto o las relaciones entre ellos.

Selección del lenguaje de modelado

Se decidió usar UML* para la modelación del sistema por ser un lenguaje de representación visual que permite combinar diversos elementos gráficos y crear diagramas, además de ser el más utilizado en el mundo para modelar sistemas que usan tecnología orientada a objetos.

1.5.6 Herramienta CASE

Según varios autores, “*Computer Aided Software Engineering*” o “*Computer Aided Systems Engineering*” (CASE*), es:

- ✓ “El *software* que se utiliza en cualquiera o en todas las fases de desarrollo de un Sistema de Información, incluyendo el análisis, diseño y programación. Las herramientas CASE* proporcionan métodos automáticos para diseñar y documentar las técnicas tradicionales de programación estructurada”. (Freedman, 1993)
- ✓ “Un amplio rango de herramientas para soportar el desarrollo de *software* han sido desarrolladas y el término *computer-aided software engineering* (CASE*) está siendo usado como un término genérico para el soporte automatizado de la ingeniería de *software*”.

Teniendo en cuenta los criterios anteriores se puede decir, que las herramientas CASE* son programas que facilitan la automatización de las diferentes fases de desarrollo de un *software* durante su ciclo de vida.

➤ *Visual Paradigm*

“*Visual Paradigm*” es una herramienta que da soporte al modelado visual, facilitando la realización de diagramas que siguen el estándar de UML* y ofreciendo:

- ✓ Entorno de creación de diagramas para UML* 2.0.
- ✓ Diseño centrado en casos de uso y enfocado al negocio que generan un *software* de mayor calidad.
- ✓ Uso de un lenguaje estándar, común a todo el equipo de desarrollo que facilita la comunicación.

Capítulo 1: *Fundamentación teórica*

- ✓ Capacidades de ingeniería directa (versión profesional) e inversa.
- ✓ Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- ✓ Disponibilidad de múltiples versiones, para cada necesidad.
- ✓ Disponibilidad de integrarse en los principales IDEs.
- ✓ Disponibilidad en múltiples plataformas.

➤ **Enterprise Architect (EA)**

“*Enterprise Architect*” es una herramienta CASE* multiusuario, está diseñada para construir *software* robustos y duraderos, provee soporte para pruebas, mantenimiento y control de cambios. EA soporta todo el proceso de UML* en un ambiente fácil de usar, rápido y flexible.

Características principales:

- ✓ Intuitivo y simple de usar.
- ✓ Bajo costo de licencia.
- ✓ Permite tanto modelar en UML* 2.0, como rastrear la información importante de proyecto con artefactos de diseño.
- ✓ Modelos de Caso de Uso, Lógicos, Dinámicos y Físicos.
- ✓ Ayuda a manejar la complejidad con instrumentos para rastrear dependencias y soporte a modelos muy extensos.
- ✓ Modelamiento de datos, Ingeniería Directa de Bases de Datos hacia DDL e Ingeniería Reversa de Bases de Datos desde ODBC.
- ✓ Corrector Ortográfico.
- ✓ Ingeniería de Código directa e inversa (sólo Ediciones Profesional y Corporativa) - Soporte para Java, C#, C ++, VB.Net, Delphi, *Visual Basic* y PHP*.
- ✓ Soporta transformaciones de Modelo avanzado de Arquitectura Conducida (MDA) usando plantillas fáciles de desarrollar y editar.
- ✓ Extensiones personalizadas para modelamiento de procesos y demás. (22)

➤ **Rational Rose**

“*Rational Rose*” es la herramienta CASE* que comercializan los desarrolladores de UML* y que soporta de forma completa la especificación del UML*. Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas, creando de esta forma un modelo completo que representa el dominio del problema y el sistema de *software*.

Capítulo 1: *Fundamentación teórica*

Selección de la herramienta CASE*

Se decidió usar como herramienta CASE* a *Visual Paradigm* porque es una herramienta de *software* libre que además de utilizar UML* como lenguaje de modelaje profesional, soporta el ciclo de vida completo del desarrollo de *software*, permite realizar ingeniería tanto directa como inversa; cabe destacar igualmente su robustez, usabilidad y portabilidad. *Visual paradigm* no es indispensable para el desarrollo del sistema pero es una herramienta muy a tener en cuenta a la hora de desarrollar un proyecto importante ya que con la realización de algunos de sus artefactos hace más entendible el diseño de la aplicación.

1.5.7 Framework (Marco de trabajo)

Un “*framework*” es una estructura *software* compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

- ✓ Un “*framework*” simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (23)

➤ *Symfony*

Symfony es un *framework* PHP* que facilita el desarrollo de las aplicaciones web, ayudando a mejorar la productividad, el uso de las buenas prácticas y de los patrones de diseño que se han definido para la web. Además, de ser el *framework* más documentado del mundo, ya que cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Características:

- ✓ Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- ✓ Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL* Server)
- ✓ Compatible solamente con PHP* 5 desde hace años, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP*.

Capítulo 1: *Fundamentación teórica*

- ✓ Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✓ Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- ✓ Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins⁷.
- ✓ Publicado bajo licencia MIT de *software* libre y apoyado por una empresa comprometida con su desarrollo.
- ✓ Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma. (24)

➤ **Zend framework**

Zend Framework se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP*, brinda soluciones para construir sitios web modernos, robustos y seguros. Además, es "Open Source" y trabaja con PHP* 5 a diferencia de CakePHP* que trabaja con PHP* 4 y PHP* 5.

Características:

- ✓ Trabaja con MVC* (Modelo Vista Controlador).
- ✓ Cuenta con módulos para manejar archivos PDF, canales RSS, Web Services (Amazon, Flickr, Yahoo), etc.
- ✓ El Marco de *Zend* también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL*.
- ✓ Completa documentación y tests de alta calidad.
- ✓ Robustas clases para autenticación y filtrado de entrada.
- ✓ Clientes para servicios web, incluidos Google Data APIs y Strikelron. (25)

➤ **Cake PHP***

Cake PHP* es un framework que ofrece una arquitectura extensible para el desarrollo, mantenimiento, e implementación de aplicaciones. Está diseñado para facilitar el desarrollo de proyectos en php*, usa los patrones de diseño de MVC* dentro de la convención sobre el paradigma de configuración, reduce los costes de desarrollo y ayuda a los programadores a escribir menos código. Además, facilita la creación de administradores para el manejo de contenido estándar (insertar, editar y eliminar registros),

⁷ Plugins: son programas que expanden las características de programas principales como el browser y le agregan capacidades multimedia. Se "conectan" a una aplicación y corren como parte de esa aplicación.

Capítulo 1: *Fundamentación teórica*

facilita la validación de los campos de los formularios, previene de ataques por inyecciones SQL*, facilita el testeo de las aplicaciones, facilita la interacción con la base de datos mediante el uso de registro activo y también tiene un sistema de plantillas rápido y flexible.

Características más destacadas:

- ✓ Arquitectura MVC* (Modelo Vista Controlador).
- ✓ *Helpers* para AJAX, JavaScript, formularios HTML* y más.
- ✓ Validación nativa.
- ✓ El código para CRUD (crear, leer, actualizar y eliminar) se llevan a cabo a través de CakePHP.
- ✓ Desarrollo rápido – Construye aplicaciones más rápido que antes.
- ✓ Buenas prácticas – Cake es muy fácil de entender y cumple los estándares en seguridad y autenticación, manejo de sesiones y muchas otras características.
- ✓ Orientado a objetos – Si te gusta la programación orientada a objetos que bien y si eres principiante te sentirás cómodo.
- ✓ Cero configuraciones – Solamente pon la información de la base de datos y la magia comenzará. (26)

Selección del *framework* a utilizar

Se decidió utilizar el *framework* *Symfony* porque posee gran cantidad de documentación disponible, es uno de los más utilizados en el mundo, incluye varias herramientas que permiten automatizar las tareas más comunes de la ingeniería del *software* y es recomendable para desarrollar aplicaciones webs complejas en las que es importante la productividad de su desarrollo, la facilidad de su mantenimiento y su seguridad.

1.6 Conclusiones

Durante el desarrollo de este capítulo:

- Se argumenta la necesidad de implementar un sistema para la Dirección de Producción Audiovisual, pues ninguno de los estudiados se ajusta parcial o totalmente a los requisitos que necesita la entidad.
- Se seleccionan las herramientas, lenguajes, metodologías y tecnologías que más se adaptan a los requisitos requeridos por el sistema a realizar.

Capítulo 2: Descripción de la solución propuesta

Capítulo 2

2.1 Introducción

Antes de construir un software, es necesario que los desarrolladores y los clientes lleguen a un entendimiento común de qué es lo que quieren que el sistema realice y las actividades que serán automatizadas. El objetivo de este capítulo, es explicar cómo funcionaría en el sistema los procesos de control y seguimiento de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual, haciendo hincapié en la planificación de las iteraciones, el plan de entrega y la fase inicial del proceso de ingeniería de *software*.

2.2 Descripción de la solución propuesta

El sistema a diseñar, debe permitir gestionar incidencias técnicas y de seguridad, mediante una interfaz gráfica, sencilla y amigable. Además, debe ser capaz de mostrar en una interfaz, el reporte de todas las incidencias efectuadas dada una fecha determinada, así como la posibilidad de imprimir dicho reporte.

Este proceso funcionará de la siguiente manera:

- ✓ En la entidad existen 6 grupos de trabajo: Control de Calidad, Seguridad Informática y Redes, Creativo, Operativo, Transcripción y Traducción, y Tecnología, Mantenimiento y Soporte.
- ✓ Cuando en un local se detecta algún incidente técnico o de seguridad, se le informa de inmediato al Jefe de Grupo correspondiente a dicho local. Si el incidente es de seguridad, entonces se reporta al grupo de seguridad informática de la entidad y el jefe de este grupo lo reporta a los técnicos en cuestión, el jefe de este grupo podrá insertar, modificar, eliminar y mostrar incidencias de este tipo en la aplicación. Los incidentes técnicos por el contrario, son reportados al grupo de tecnología, mantenimiento y soporte y el jefe de este grupo podrá insertar, modificar, eliminar y mostrar incidencias técnicas en la aplicación, así como informarle a sus técnicos de la incidencia.
- ✓ Los técnicos realizan un diagnóstico más preciso, con el objetivo de determinar las posibles causas de la incidencia para solucionarla: si le dieron solución, entonces lo registrarán en el sistema. Las incidencias técnicas, que no puedan ser solucionadas por estos técnicos, se reportan al grupo de soporte técnico de la universidad.
- ✓ Para saber la existencia de alguna incidencia u obtener algún reporte, se procede a realizar una búsqueda en la aplicación. Tanto los Jefes de grupo como el Jefe del Departamento y el Director, podrán realizar estas actividades.

Capítulo 2: *D*escripción de la solución propuesta

2.3 Especificación de requisitos

Para dar cumplimiento al objetivo planteado y conocidos ya los conceptos que rodean al objeto de estudio, se deben definir una serie de requisitos que permitan resolver el problema existente.

➤ **Requerimientos funcionales**

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Por lo general se describen mejor a través del modelo de casos de uso (MCU) y los casos de uso (CU) como tal. Por lo tanto, los requerimientos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. (27)

RF1. Autenticar Usuario

RF 1.1 Iniciar sesión.

RF 1.2 Cerrar sesión.

RF2. Gestionar Usuario

2.1 Adicionar usuario.

2.2 Eliminar usuario.

2.3 Modificar usuario.

2.4 Mostrar usuario.

RF3. Mostrar Permiso

RF4. Gestionar incidencia técnica

RF4.1 Adicionar incidencia técnica.

RF4.2 Modificar incidencia técnica.

RF4.3 Mostrar incidencia técnica.

RF4.4 Eliminar incidencia técnica.

RF5. Gestionar incidencia seguridad

RF5.1 Adicionar incidencia seguridad.

RF5.2 Modificar incidencia seguridad.

RF5.3 Mostrar incidencia seguridad.

RF5.4 Eliminar incidencia seguridad.

Capítulo 2: Descripción de la solución propuesta

RF6: Gestionar soluciones incidencias técnicas

RF6.1 Adicionar soluciones incidencias técnicas.

RF6.2 Modificar soluciones incidencias técnicas.

RF6.3 Mostrar soluciones incidencias técnicas.

RF6.4 Eliminar soluciones incidencias técnicas.

RF7 Gestionar soluciones incidencias seguridad

RF7.1 Adicionar soluciones incidencias seguridad.

RF7.2 Modificar soluciones incidencias seguridad.

RF7.3 Mostrar soluciones incidencias seguridad.

RF7.4 Eliminar soluciones incidencias seguridad.

RF8. Generar listado reportes incidencias

RF9. Buscar Incidencias técnicas

RF10. Buscar Incidencias seguridad

➤ **Requerimientos no funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente, están vinculados a requerimientos funcionales, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Teniendo en cuenta las necesidades del cliente, así como el nivel tecnológico que posee el mismo, y las características del entorno en que se desarrolla esta aplicación, se definen las propiedades o cualidades que dicho sistema debe poseer. (27)

Usabilidad

RNF 1. El sistema debe estar disponible las veinticuatro horas del día, sin ninguna interrupción.

RNF 2. El sistema debe ser accesible desde todos los puntos donde exista una máquina conectada a la red.

RNF 3. El sistema debe ser una interfaz de fácil aprendizaje para que usuarios inexpertos puedan familiarizarse lo más pronto posible y le sea cómodo el manejo del *software*.

Capítulo 2: Descripción de la solución propuesta

RNF 4. El sistema debe diferenciar las interfaces gráficas y opciones para los usuarios que accedan al sistema con diferentes roles.

Rendimiento

RNF 5. El sistema debe ser capaz de mantener el mismo rendimiento y estabilidad a medida que aumenta la cantidad de datos a gestionar.

RNF 6. Al ser montado el sistema debe ser capaz de correr todos sus componentes tales como Gestor de Bases de Datos y Aplicación Web, en una misma PC.

Soporte

RNF 7. Es necesario que el producto reciba mantenimiento ante los posibles fallos que puedan surgir.

Portabilidad

RNF 8. El sistema funcionará sobre plataforma Windows, en caso de cambio de plataforma por pedidos del cliente, el sistema se ajustará totalmente con la mayor brevedad.

RNF 9. El producto está implantado sobre una plataforma Web, codificada en PHP* 5.0 y su sistema de base de datos en MySQL.

Seguridad

RNF 10. *Confidencialidad:* La información manejada por el sistema contará de protección ante intrusos y accesos no autorizados, será vista únicamente por aquellos usuarios que tengan derecho a verla.

RNF 12. *Integridad:* La información manejada por el sistema debe ser objeto de cuidadosa protección contra la corrupción y estados de inconsistencia.

RNF 13. *Disponibilidad:* La aplicación deberá estar disponible en todo momento para aquellas personas con acceso a la información, y los mecanismos utilizados para lograr la seguridad, no deben ser un obstáculo a los usuarios para obtener los datos deseados en un momento dado.

Legales

RNF 14. El sistema es reconocido y autorizado por la entidad.

Requerimientos de *software*

RNF 15. En las computadoras de los usuarios solo se requiere un navegador Web.

RNF 16. El sistema operativo debe ser Windows NT en adelante o cualquier distribución de Linux.

RNF 17. La computadora servidora debe tener instalado el WampServer.

Capítulo 2: Descripción de la solución propuesta

Requerimientos de hardware

RNF 18. Las PCs implicadas en el funcionamiento de la aplicación deben estar conectadas a la red.

RNF 19. Las computadoras clientes requieren 128 MB de RAM como mínimo.

RNF 20. El servidor Web junto con el servidor de base de datos deben tener 256 MB de RAM y 20GB de disco duro como mínimo.

2.4 Personas relacionadas con el sistema

Se define como persona relacionada con el sistema, a toda aquella que obtiene un resultado del valor de uno o varios procesos que se ejecutan. Además de aquellas que se encuentran involucradas en dichos procesos, pues participan en ellos pero no obtienen ningún resultado de valor.

Personas	Descripción
Usuario	Se encarga de interactuar con el sistema, generaliza el acceso al mismo y tiene que autenticarse antes de entrar.
Directivo (Jefe de Departamento y Director)	Realiza búsquedas de incidencias. Así como generar un reporte.
Jefe de Grupo de seguridad informática	Es el encargado de gestionar las incidencias de seguridad en el sistema, así como realizar búsquedas de incidencias de seguridad.
Jefe de Grupo de tecnología, mantenimiento y soporte	Es el encargado de gestionar las incidencias técnicas en el sistema, así como realizar búsquedas de incidencias técnicas.
Administrador del sistema	Es el encargado de darles los permisos a los usuarios en la aplicación y de gestionar usuarios del sistema. El mismo tendrá la potestad de realizar alguna que otra actividad en el sistema de ser necesario, ya que es el que administra el sistema completo.
Técnico de mantenimiento y soporte	Es el encargado de gestionar en el sistema la solución del incidente técnico ocurrido.
Técnico de seguridad informática	Es el encargado de gestionar en el sistema la solución del incidente de seguridad ocurrido.

Tabla # 2.1 Personas relacionadas con el sistema.

Capítulo 2: Descripción de la solución propuesta

2.5 Fase de Planificación

La metodología de Programación Extrema (XP*) tiene 4 fases fundamentales: Planificación, Diseño, Desarrollo y Prueba; en el presente capítulo solo se hace referencia a la primera fase.

XP* plantea la planificación como un diálogo entre el desarrollador y el cliente, los que llegarán a un consenso con respecto al alcance del proyecto, la prioridad a la hora de implementar las funcionalidades requeridas, la composición de las versiones y la fecha de entrega de las mismas.

Los principales artefactos de esta fase son las historias de usuario, que sirven de guía a todo el proceso de desarrollo.

2.5.1 Historias de Usuarios (HU*)

Las historias de usuario son utilizadas para especificar los requisitos del *software* y las escriben los propios clientes según su percepción de las necesidades del sistema. Proporcionan los detalles sobre la estimación del riesgo y tiempo que llevará la implementación de determinadas funcionalidades. Son una vista a muy alto nivel del costo y esfuerzo del trabajo que se desea desarrollar. Deben poder ser programadas en un tiempo entre una y tres semanas. Las historias de usuarios a implementar son:

➤ Autenticar

Permite la autenticación de usuarios en el sistema, con el objetivo de que accedan sólo los usuarios con permiso para hacerlo. Restringe las partes del sistema que pueden ser accedidas por los diferentes usuarios autenticados. Esto proporciona que la información que se almacena en el sistema sólo pueda ser consultada por personas que tengan permiso para hacerlo.

Historia de Usuario	
Número: 1	Nombre: Autenticar
Usuario: Jefe de Grupo de seguridad informática, Jefe de Grupo de tecnología, mantenimiento y soporte, Jefe de Departamento, Director y Técnicos.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 0.5	Iteración Asignada: 1
Descripción: Solicitar un nombre de usuario y una contraseña para acceder a la aplicación en dependencia del rol que represente el usuario.	
Observaciones: Si el usuario o la contraseña son incorrectos, tiene la posibilidad de introducirlos nuevamente.	

Tabla # 2.2 HU* Autenticar.

Capítulo 2: Descripción de la solución propuesta

➤ Gestionar Usuarios

El administrador del sistema será el encargado de adicionar, modificar, mostrar y eliminar usuarios al sistema.

Historia de Usuario	
Número: 2	Nombre: Gestionar Usuario
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realizan las acciones de insertar, modificar, mostrar y eliminar.	
Observaciones:	

Tabla #2.3 HU* Gestionar Usuario.

➤ Mostrar Permisos

El administrador del sistema podrá ver los permisos de los usuarios.

Historia de Usuario	
Número: 3	Nombre: Mostrar Permiso
Usuario: Administrador	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: Se realiza la acción de mostrar permiso	
Observaciones:	

Tabla #2.4 HU* Mostrar permiso.

➤ Gestionar incidencias técnicas

Permite adicionar, mostrar, modificar y eliminar los datos de cada incidencia técnica, con el objetivo de mantenerlas registradas y de ser necesario, modificarlas o mostrarlas en algún momento. Esta funcionalidad la realiza el Jefe de grupo de tecnología -mantenimiento y soporte.

Historia de Usuario	
Número: 4	Nombre: Gestionar incidencias técnicas
Usuario: Jefe de Grupo de tecnología -mantenimiento y soporte.	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1.5	Iteración Asignada: 1
Descripción: Se realizan las acciones de adicionar, mostrar, modificar y eliminar	
Observaciones:	

Tabla # 2.5 HU* Gestionar incidencias técnicas.

Capítulo 2: Descripción de la solución propuesta

➤ Gestionar incidencias de seguridad

Permite adicionar, mostrar, modificar y eliminar las incidencias de seguridad, con el propósito de mantenerlas almacenadas en el sistema y de ser necesario, modificarlas o mostrarlas en algún momento. Esta funcionalidad la realiza el Jefe de grupo de seguridad informática.

Historia de Usuario	
Número: 5	Nombre: Gestionar incidencias de seguridad
Usuario: Jefe de Grupo de seguridad informática	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1.5	Iteración Asignada: 1
Descripción: Se realizan las acciones de adicionar, mostrar, modificar y eliminar	
Observaciones:	

Tabla # 2.6 HU* Gestionar incidencias seguridad.

➤ Gestionar soluciones incidencias técnicas

Permite adicionar, mostrar, modificar y eliminar las soluciones que les dé el técnico de mantenimiento y soporte a las incidencias. Esta funcionalidad es muy importante, ya que la información guardada, en otro momento puede ser de gran utilidad.

Historia de Usuario	
Número: 6	Nombre: Gestionar soluciones a incidencias técnicas
Usuario: Técnico de mantenimiento y soporte.	
Prioridad en negocio: Media	Riesgo en Desarrollo: Alta
Puntos de Estimación: 0.5	Iteración Asignada: 2
Descripción: Se registra si se le dio solución o no a la incidencia.	
Observaciones:	

Tabla # 2.7 HU* Gestionar soluciones incidencias técnicas

➤ Gestionar soluciones incidencias de seguridad

Permite adicionar, mostrar, modificar y eliminar las soluciones que les dé el técnico de seguridad informática a las incidencias. Esta funcionalidad es muy importante, ya que la información guardada, en otro momento puede ser de gran utilidad.

Capítulo 2: Descripción de la solución propuesta

Historia de Usuario	
Número: 7	Nombre: Gestionar soluciones a incidencias de seguridad
Usuario: Técnico de seguridad informática.	
Prioridad en negocio: Media	Riesgo en Desarrollo: Alto
Puntos de Estimación: 0.5	Iteración Asignada: 2
Descripción: Se registra si se lo dio solución o no a la incidencia.	
Observaciones:	

Tabla # 2.8 HU* Gestionar soluciones incidencias seguridad

➤ Generar listado de reportes de incidencias

Permite la elaboración del reporte “Incidencias de Seguridad y Técnicas”. Mediante la selección, el directivo puede seleccionar la fecha en el que se desee obtener el reporte.

Historia de Usuario	
Número: 8	Nombre: Generar listado de reportes de incidencias.
Usuario: Director, Jefe de Departamento.	
Prioridad en negocio: Baja	Riesgo en Desarrollo: Bajo
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: Se muestra un listado con los reportes de las incidencias hechas en un rango de tiempo.	
Observaciones:	

Tabla # 2.9 HU* Generar listado reportes incidencias.

➤ Buscar Incidencias técnicas

Permite la búsqueda según parámetros. Los parámetros son entrados por el director, el jefe de departamento y el jefe de grupo de tecnología - mantenimiento y soporte.

Historia de Usuario	
Número: 9	Nombre: Buscar Incidencias técnicas
Usuario: Director, Jefe de Grupo de tecnología mantenimiento y soporte, Jefe de Departamento	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 0.5	Iteración Asignada: 1
Descripción: Se seleccionan las opciones de búsqueda y el sistema muestra la incidencia.	
Observaciones:	

Tabla # 2.10 HU* Buscar Incidencias técnicas

Capítulo 2: Descripción de la solución propuesta

➤ Buscar Incidencias de seguridad

Permite la búsqueda según parámetros. Los parámetros son entrados por el director, el jefe de departamento y el jefe de grupo de seguridad informática.

Historia de Usuario	
Número: 10	Nombre: Buscar Incidencias de seguridad
Usuario: Director, Jefe de Grupo de seguridad informática, Jefe de Departamento	
Prioridad en negocio: Alta	Riesgo en Desarrollo: Alto
Puntos de Estimación: 0.5	Iteración Asignada: 1
Descripción: Se seleccionan las opciones de búsqueda y el sistema muestra la incidencia.	
Observaciones:	

Tabla # 2.11 HU* Buscar Incidencias seguridad

2.5.2 Estimación de esfuerzo por HU*

Las estimaciones de esfuerzo asociado a la implementación de las historias de usuario, la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación, donde los miembros de los equipos de desarrollo trabajan el tiempo planeado sin ningún tipo de interrupción. Esta estimación incluye todo el esfuerzo asociado a la implementación de la historia de usuario. Para el logro del sistema se ha realizado la estimación de esfuerzo por historias de usuarios, estas se muestran a continuación:

Historias de Usuario	Puntos de estimación
Autenticar	0.5
Gestionar usuario	1
Mostrar permiso	1
Gestionar incidencias técnicas	1.5
Gestionar incidencias de seguridad	1.5
Buscar incidencias técnicas	0.5
Buscar incidencias de seguridad	0.5
Gestionar soluciones a incidencias técnicas	0.5
Gestionar soluciones a incidencias de seguridad	0.5
Generar listado de reportes de incidencias	1

Tabla # 2.12 Estimación de esfuerzo por HU*

2.5.3 Plan de iteraciones

El plan de iteraciones no es más que las partes en las que se divide la implementación.

Iteración 1: En esta iteración se implementan las historias de usuario de mayor prioridad; las cuales

Capítulo 2: Descripción de la solución propuesta

nos brinda la posibilidad de buscar incidencias, gestionar usuarios, mostrar permisos, gestionar incidencias técnicas y de seguridad. Además, con esta iteración el usuario puede autenticarse. De esta manera, se trata de tener preparadas las funcionalidades básicas e indispensables del sistema. También se tendrá la primera versión de pruebas.

Iteración 2: En esta iteración se implementan las funcionalidades con prioridad media, las cuales permiten gestionar las soluciones de las incidencias técnicas y las de seguridad. Al cliente se le mostrará las versiones de pruebas anteriores junto a las correspondientes a esta iteración.

Iteración 3: En esta iteración se implementan las historias de usuario de baja prioridad. Estas funcionalidades brindan al sistema cierta comodidad en la gestión de las funcionalidades de prioridad alta, además de corregir errores anteriores. Esta iteración permitirá generar un listado con las incidencias reportadas. A partir de este momento quedará conformado totalmente el sistema y sometido a pruebas rigurosamente.

2.5.4 Plan de duración de iteraciones

El plan de duración de iteraciones tiene como finalidad mostrar la duración de cada iteración y el orden en que se implementan las historias de usuario, teniendo una mayor organización.

Iteraciones	Orden de las historias de usuario a implementar	Duración total de la iteración
Iteración 1	Autenticar Gestionar usuario Mostrar Permisos Gestionar incidencias técnicas Gestionar incidencias de seguridad Buscar incidencias técnicas Buscar incidencias seguridad	6.5 semanas
Iteración 2	Gestionar soluciones incidencias técnicas Gestionar soluciones incidencias seguridad	1 semana
Iteración 3	Generar listado reportes incidencias	1 semana

Tabla # 2.13 Plan de duración de iteraciones

Capítulo 2: Descripción de la solución propuesta

2.5.5 Plan de entregas

Con el plan de duración se da un aproximado de las entregas de las versiones, contando que la implementación empieza en el mes de abril.

A entregar	Final 1ra Iteración (22/04/2010)	Final 2da Iteración (25/04/2010)	Final 3ra Iteración (06/05/2010)
Sistema de gestión de incidencias técnicas y de seguridad	0.1	0.2	1.0

Tabla # 2.14 Plan de entregas

2.6 Conclusiones

Al finalizar este capítulo se obtiene como resultado:

- La propuesta del sistema, detallando como ocurren los procesos que se llevan a cabo en la gestión de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.
- Los requerimientos funcionales y no funcionales del sistema, que dan respuesta al problema planteado en la investigación.
- Las historias de usuario, el plan de iteraciones, el plan de duración de iteraciones y el plan de entrega, que constituyen la base para la implementación del sistema.
- La estimación del tiempo para implementar el sistema, teniendo en cuenta las historias de usuario.

Capítulo 3: Construcción de la solución propuesta

Capítulo 3

3.1 Introducción

En este capítulo se describe el proceso de diseño de la aplicación a través de varios artefactos propuestos por la metodología XP*. Como exige dicha metodología, la herramienta ha sido implementada de forma iterativa, obteniendo al culminar de cada iteración, un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión del desarrollador con la opinión de este último. Además, se detallan las tres iteraciones llevadas a cabo durante la etapa de construcción del sistema, exponiéndose fundamentalmente las Tarjetas CRC y el modelado en lenguaje UML* de la arquitectura del sistema, las tareas generadas por cada historia de usuario, así como las pruebas de aceptación efectuadas sobre el sistema.

3.2 Arquitectura del sistema

En la construcción de la aplicación se utiliza una arquitectura cliente/servidor. En esta arquitectura las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario. De esta manera, cuando el cliente envía un mensaje solicitando un determinado servicio a un servidor, estos últimos responden a la demanda del cliente que la produjo.

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente. (28)

Ventajas de la seguridad del sistema:

- El sistema permite autenticarse a través de un Módulo de administración.
- El tiempo entre fallos es breve o cero, haciendo lo posible para que esto no ocurra.
- El servicio del sistema tiene una disponibilidad aceptable (99%).
- El sistema tiene la capacidad de darle seguridad al usuario, de que las informaciones solo serán vistas por quien esté capacitado para esto.
- El sistema tiene la capacidad de identificar con certeza a los diversos usuarios o entidades que interactúan con el.

Capítulo 3: Construcción de la solución propuesta

- La base de datos está fraccionada sobre varios esquemas, dividiendo así de una forma lógica las funcionalidades, evitando la pérdida total de la información en caso de algún accidente o ataque.

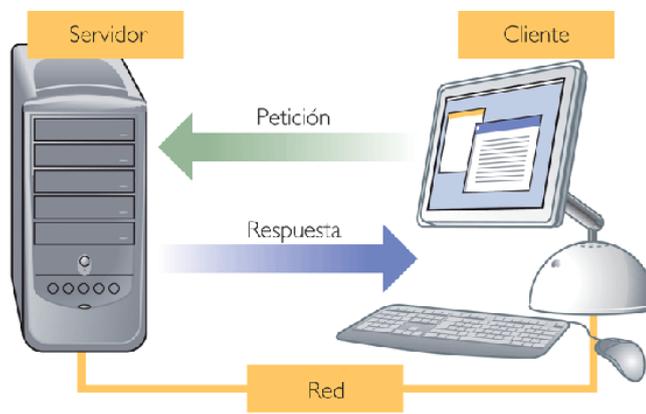


Figura #2 Arquitectura Cliente – Servidor

3.3 Fase de diseño

3.3.1 Patrones de diseño

- **Modelo Vista Controlador**

El patrón de diseño MVC* define una manera de organizar el código de acuerdo con su naturaleza. Este patrón separa el código en tres capas:

- ✓ La capa **Modelo** define la lógica de negocio (la base de datos pertenece a esta capa). *Symfony* guarda todas las clases y archivos relacionados con el modelo en el directorio `lib/model/`.
- ✓ La **Vista** es con lo que el usuario interactúa. En *Symfony*, la vista es principalmente la capa de plantillas PHP*. Estas son guardadas en el directorio `templates/`.
- ✓ El **Controlador** es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente. Todas las solicitudes son gestionadas por un controlador frontal (`index.php` y `Aplicación_dev.php`). Estos controladores frontales delegan la verdadera labor a las acciones, estas acciones son, lógicamente, agrupadas en módulos.

El patrón representa un mecanismo de mejora de procesos de desarrollo de software, fácil de comprender y aplicar. Permite obtener un sistema claro y bien especificado. La separación de la interfaz del resto del código eliminará problemas como:

- ✓ Que a la complejidad de los cálculos del programa se le suma la complejidad de la interfaz de usuario, lo que proporcionaría un código de mayor calidad y de menor dificultad.
- ✓ Menor rigidez en la interfaz con el usuario, dado que el código HTML no estaría entremezclado con el lenguaje de programación y sería una tarea más fácil cambiar el diseño de la interfaz.

Capítulo 3: Construcción de la solución propuesta

➤ Patrones de diseño utilizados en *Symfony*

En el sistema se puede observar que cada una de las páginas tiene el mismo aspecto; para esto se utiliza el patrón decorator, donde la plantilla es decorada después de que el contenido es mostrado por una plantilla global, llamada *layout* en *Symfony*.

El *layout* de una aplicación se puede encontrar en el directorio `apps/Aplicación/templates/` y se llama `layout.php`. Este directorio contiene todas las plantillas globales para una aplicación.

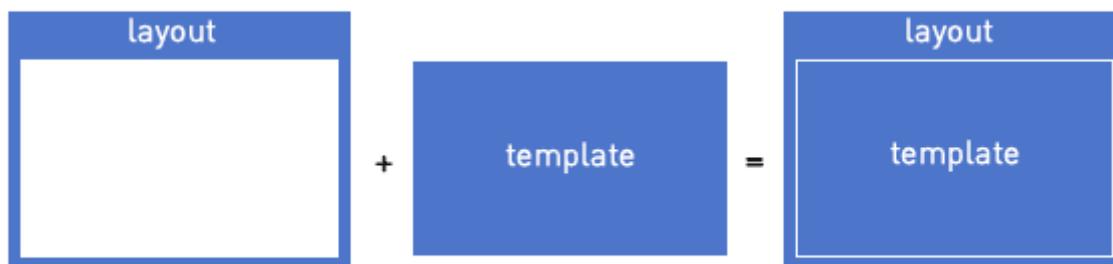


Figura # 3 Plantilla global en *Symfony*.

3.3.2 Tarjetas CRC

Las tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración) son técnicas de modelado creadas para ayudar a los desarrolladores de *software*, a crear diseños de clases orientados a responsabilidades. La metodología XP* estipula el uso de las tarjetas CRC como un artefacto obligatorio durante el desarrollo de un proyecto.

¿En qué consisten las tarjetas CRC? Pues en eso, en tarjetas (papel o cartón) de aproximadamente 10x15 cm. En ellas, se diferencian tres zonas: el nombre de la clase, sus responsabilidades y los colaboradores. (29)

IdapEmpet	
Responsabilidad	Colaboraciones (HU*)
Iniciar sección Cerrar sección	

Tabla # 3.1 Tarjeta CRC: Clase *IdapEmpet*

Capítulo 3: Construcción de la solución propuesta

sf_guard_user	
Responsabilidad	Colaboraciones
Adicionar usuario Modificar usuario Mostrar usuario Eliminar usuario	IdapEmpet

Tabla # 3.2 Tarjeta CRC: Clase sf_guard_user

sf_guard_permission	
Responsabilidad	Colaboraciones
Mostrar permiso	IdapEmpet sf_guard_user

Tabla # 3.3 Tarjeta CRC: Clase sf_guard_permission

crudIncidenciaT	
Responsabilidad	Colaboraciones
Adicionar incidencia técnica Modificar incidencia técnica Mostrar incidencia técnica Eliminar incidencia técnica	IdapEmpet sf_guard_user sf_guard_permission

Tabla # 3.4 Tarjeta CRC: Clase crudIncidenciaT

crudIncidenciaS	
Responsabilidad	Colaboraciones
Adicionar incidencia seguridad Modificar incidencia seguridad Mostrar incidencia seguridad Eliminar incidencia seguridad	IdapEmpet sf_guard_user sf_guard_permission

Tabla # 3.5 Tarjeta CRC Clase: crudIncidenciaS

Capítulo 3: Construcción de la solución propuesta

crudSolucionIt	
Responsabilidad	Colaboraciones
Adicionar solución incidencia técnica	IdapEmpet
Modificar solución incidencia técnica	sf_guard_user
Mostrar solución incidencia técnica	sf_guard_permission
Eliminar solución incidencia técnica	crudIncidenciaT

Tabla # 3.6 Tarjeta CRC Clase: crudSolucionIt

crudSolucionIs			
Responsabilidad			Colaboraciones
Adicionar	solución	incidencia	IdapEmpet sf_guard_user
seguridad			sf_guard_permission
Modificar	solución	incidencia	crudIncidenciaS
seguridad			
Mostrar	solución	incidencia	
seguridad			
Eliminar	solución	incidencia	
seguridad			

Tabla # 3.7 Tarjeta CRC Clase: crudSolucionIs

Reporte	
Responsabilidad	Colaboraciones
Generar reportes	IdapEmpet
	sf_guard_user
	sf_guard_permission
	crudIncidenciaT
	crudIncidenciaS

Tabla # 3.8 Tarjeta CRC Clase: reportes incidencias

Capítulo 3: Construcción de la solución propuesta

buscar_incidenciaT	
Responsabilidad	Colaboraciones
Buscar incidencia técnica	IdapEmpet sf_guard_user sf_guard_permission crudIncidenciaT

Tabla # 3.9 Tarjeta Clase: buscar_incidenciaT

buscar_incidenciaS	
Responsabilidad	Colaboraciones
Buscar incidencia seguridad	IdapEmpet sf_guard_user sf_guard_permission crudIncidenciaS

Tabla # 3.10 Tarjeta Clase: buscar_incidenciaS

3.3.3 Diseño de la Base de Datos

Una de las tareas más importantes a la hora de construir una aplicación Web, es la base de datos, ya que uno de sus objetivos fundamentales es brindar la persistencia al modelo que se describe en el capítulo. En esta investigación fueron construidos dos modelos para la representación de los datos persistentes: el Modelo Lógico de Datos y el Modelo Físico de Datos.

3.4 Fase de codificación

3.4.1 Tareas de la ingeniería

Durante el transcurso de las iteraciones se realiza la implementación de las historias de usuario que fueron seleccionadas en cada una de ellas. Al principio se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte del plan se descomponen las historias en tareas de ingeniería.

Para definir las tareas de la ingeniería se cuenta con una plantilla que permite definir cada una de las actividades a las que están asociadas las historias de usuario y que permiten su implementación. A continuación se muestran las tareas asignadas por iteraciones.

Iteración 1

En esta iteración se implementan 7 historias de usuario que proveen la parte esencial de la aplicación.

Capítulo 3: Construcción de la solución propuesta

Historias de Usuario	Tiempo de Implementación	
	Estimación	Tiempo Real
Autenticar	0.5	0.5
Gestionar Usuario	1	1
Mostrar Permiso	1	1
Gestionar incidencias técnicas	1.5	1.5
Gestionar incidencias de seguridad	1.5	1.5
Buscar incidencias técnicas	0.5	0.5
Buscar incidencias de seguridad	0.5	0.5

Tabla # 3.11 Historias de usuarios desarrolladas en la 1ra iteración

Tarea de Ingeniería	
Número de la Tarea: 1	Número de la HU*:
Nombre tarea: Crear interfaz del sistema	
Tipo de Tarea: Diseño	
Fecha Inicio: 11/04/2010	Fecha Fin: 11/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Se diseña la página principal del sistema	

Tabla # 3.12 Tarea para crear la interfaz del sistema

Tarea de Ingeniería	
Número de la Tarea: 2	Número de la HU*: 1
Nombre tarea: Mostrar formulario para autenticarse	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 11/04/2010	Fecha Fin: 16/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Mostrar un formulario con los datos de inicio de sección (usuario y contraseña) y el botón aceptar, para acceder al sistema.	

Tabla # 3.13 Tarea: Mostrar formulario de autenticación

Capítulo 3: Construcción de la solución propuesta

Tarea de Ingeniería	
Número de la Tarea: 3	Número de la HU*: 1
Nombre tarea: Validar campos para autenticarse	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 11/04/2010	Fecha Fin: 16/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Verificar que los datos que se inserten en los campos, son los correctos.	

Tabla # 3.14 Tarea: Validar campos para autenticarse

Tarea de Ingeniería	
Número de la Tarea: 4	Número de la HU*: 2
Nombre tarea: Diseñar el formulario para gestionar usuarios	
Tipo de Tarea: Diseño	
Fecha Inicio: 11/04/2010	Fecha Fin: 16/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Se crean y habilitan todos los campos necesarios del formulario para la gestión de usuarios.	

Tabla # 3.15 Tarea: Diseño del formulario para gestionar usuarios

Tarea de Ingeniería	
Número de la Tarea: 5	Número de la HU*: 2
Nombre tarea: Validar campos para gestionar usuarios	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 11/04/2010	Fecha Fin: 16/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Verificar que los datos que se inserten en los campos son los correctos.	

Tabla # 3.16 Tarea: Validar campos para gestionar usuarios

Capítulo 3: Construcción de la solución propuesta

Tarea de Ingeniería	
Número de la Tarea: 6	Número de la HU: 3
Nombre tarea: Diseñar el formulario para mostrar permisos	
Tipo de Tarea: Diseño	
Fecha Inicio: 11/04/2010	Fecha Fin: 16/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Se crean y habilitan todos los campos necesarios del formulario para que se muestren los permisos.	

Tabla # 3.17 Tarea: Diseño del formulario para mostrar permisos

Tarea de Ingeniería	
Número de la Tarea: 8	Número de la HU*: 4
Nombre tarea: Diseñar el formulario para gestionar incidencias técnicas	
Tipo de Tarea: Diseño	
Fecha Inicio: 18/04/2010	Fecha Inicio: 18/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Crea un formulario que permitirá insertar, modificar, mostrar y eliminar una incidencia técnica.	

Tabla # 3.18 Tarea: Diseño del formulario para gestionar incidencias técnicas

Tarea de Ingeniería	
Número de la Tarea: 9	Número de la HU*: 4
Nombre tarea: Validar campos del formulario para gestionar incidencias técnicas	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 18/04/2010	Fecha Inicio: 18/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Verificar que los datos que se inserten en los campos son los correctos.	

Tabla # 3.19 Tarea: Validar campos del formulario para gestionar incidencias técnicas

Capítulo 3: Construcción de la solución propuesta

Tarea de Ingeniería	
Número de la Tarea: 10	Número de la HU*: 5
Nombre tarea: Diseñar el formulario para gestionar incidencias de seguridad	
Tipo de Tarea: Diseño	
Fecha Inicio: 19/04/2010	Fecha Inicio: 19/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Crea un formulario que permitirá insertar, modificar, mostrar y eliminar una incidencia de seguridad.	

Tabla # 3.20 Tarea: Diseñar el formulario para gestionar incidencias de seguridad

Tarea de Ingeniería	
Número de la Tarea: 11	Número de la HU*: 5
Nombre tarea: Validar campos del formulario para gestionar incidencias de seguridad	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 19/04/2010	Fecha Inicio: 19/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Verificar que los datos que se inserten en los campos, son los correctos	

Tabla # 3.21 Tarea: Validar campos del formulario para gestionar incidencias seguridad

Tarea de Ingeniería	
Número de la Tarea: 12	Número de la HU*: 6
Nombre tarea: Buscar incidencias técnicas	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 21/04/2010	Fecha Inicio: 21/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Permite buscar una incidencia técnica introduciendo parámetros de la misma.	

Tabla # 3.22 Tarea: Buscar incidencias técnicas

Tarea de Ingeniería	
Número de la Tarea: 13	Número de la HU*: 7
Nombre tarea: Buscar incidencias seguridad	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 22/04/2010	Fecha Inicio: 22/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Permite buscar una incidencia de seguridad introduciendo parámetros de la misma.	

Tabla # 3.23 Tarea: Buscar incidencias seguridad

Capítulo 3: Construcción de la solución propuesta

Iteración 2

En esta iteración se implementan dos historias de usuario.

Historias de Usuario	Tiempo de Implementación	
	Estimación	Tiempo Real
Gestionar soluciones incidencias técnicas	0.5	1
Gestionar soluciones incidencias de seguridad.	0.5	1

Tabla # 3.24 Historias de usuarios desarrolladas en la 2da iteración

Tarea de Ingeniería	
Número de la Tarea: 14	Número de la HU*: 8
Nombre tarea: Diseñar el formulario para gestionar solución incidencias técnicas	
Tipo de Tarea: Diseño	
Fecha Inicio: 24/04/2010	Fecha Inicio: 24/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Crea un formulario que permitirá insertar, modificar, mostrar y eliminar una solución para las incidencias técnicas.	

Tabla # 3.25 Tarea: Diseñar el formulario para gestionar solución incidencias técnicas

Tarea de Ingeniería	
Número de la Tarea: 15	Número de la HU*: 8
Nombre tarea: Validar campos del formulario para gestionar solución incidencias técnicas	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 24/04/2010	Fecha Inicio: 24/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Verificar que los datos que se inserten en los campos son los correctos.	

Tabla # 3.26 Tarea: Validar campos del formulario para gestionar solución incidencias técnicas

Capítulo 3: Construcción de la solución propuesta

Tarea de Ingeniería	
Número de la Tarea: 16	Número de la HU*: 9
Nombre tarea: Diseñar el formulario para gestionar solución incidencias de seguridad	
Tipo de Tarea: Diseño	
Fecha Inicio: 25/04/2010	Fecha Inicio: 25/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Crea un formulario que permitirá insertar, modificar, mostrar y eliminar una solución para las incidencias de seguridad.	

Tabla # 3.27 Tarea: Diseñar el formulario para gestionar solución incidencias de seguridad

Tarea de Ingeniería	
Número de la Tarea: 17	Número de la HU*: 9
Nombre tarea: Validar campos del formulario para gestionar solución incidencias de seguridad	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 25/04/2010	Fecha Inicio: 25/04/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Verificar que los datos que se inserten en los campos son los correctos.	

Tabla # 3.28 Tarea: Validar campos del formulario para gestionar solución a incidencias de seguridad

Iteración 3

En esta iteración se implementó una historia de usuario. Al finalizar se cuenta con un producto listo para poner en funcionamiento.

Historias de Usuario	Tiempo de Implementación	
	Estimación	Tiempo Real
Generar listado de reportes de incidencias	1	1

Tabla # 3.29 Historias de usuarios desarrolladas en la 3ra iteración

Tarea de Ingeniería	
Número de la Tarea: 18	Número de la HU*: 10
Nombre tarea: Generar listado de incidencias	
Tipo de Tarea: Desarrollo	
Fecha Inicio: 27/04/2010	Fecha Inicio: 06/05/2010
Programador responsable: Jasiel Pavón Leyva – Yaima García Ruiz	
Descripción: Genera un listado de las incidencias dada la fecha de inicio y fin.	

Tabla # 3.30 Tarea: Generar listado de incidencias

Capítulo 3: Construcción de la solución propuesta

3.4.2 Interfaz de Usuario

Estas interfaces se encuentran en el **anexo**.

3.4.3 Diagrama de despliegue

El diagrama de despliegue permite apreciar de forma visual, cómo se encuentran relacionados físicamente los nodos de la aplicación. La aplicación se encuentra hospedada en un servidor Web y se comunica con el SGBD* MySQL.

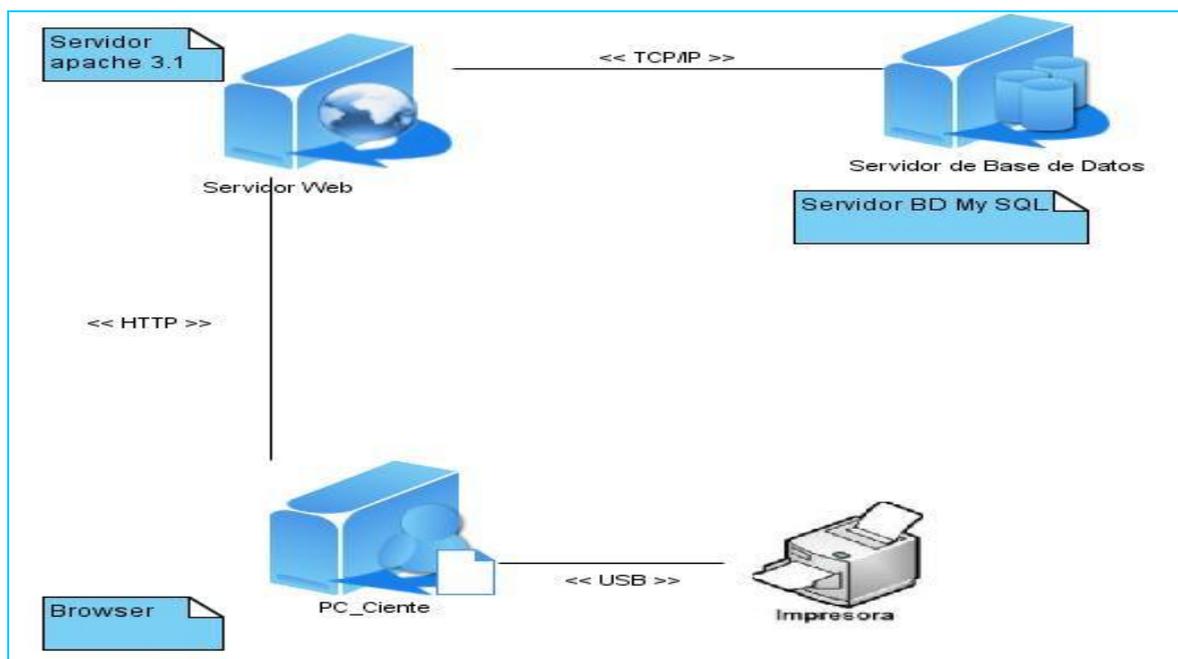


Figura # 4 Diagrama de Despliegue

3.5 Fase de pruebas

En XP* uno de los pilares más importantes es el proceso de pruebas, en el que los desarrolladores prueban tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados. Todo esto permite una mejor calidad en los productos desarrollados y la seguridad de los programadores a la hora de introducir algún cambio o modificación en el sistema.

La metodología XP* divide las pruebas en dos grupos: las pruebas unitarias, que son realizadas por los programadores, encargadas de verificar el código; y las pruebas de aceptación, destinadas a verificar si al final de cada iteración se obtuvo la funcionalidad que se requiere, además de comprobar que dicha funcionalidad sea lo que el cliente quiere. (30)

Capítulo 3: Construcción de la solución propuesta

3.5.1 Pruebas de Aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario. Durante las iteraciones, las HU* seleccionadas, serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU* ha sido implementada correctamente. Una HU* puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas pruebas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. Una HU* no se considera completa hasta que no ha pasado por sus pruebas de aceptación. (30)

3.6 Conclusiones

Al concluir el presente capítulo se obtienen resultados significativos en la investigación, ellos son:

- El diseño de la base de datos, que posibilitó un mejor entendimiento de cómo se relacionan las tablas en la base de datos.
- Las tarjetas CRC con las clases que intervienen en la solución, lo que le permitió a los programadores crear diseños de clases orientadas a responsabilidades.
- El prototipo de interfaz gráfica, que reúne las mejores características de las partes del sistema.
- Las descripciones de las pruebas de aceptación aseguraron que el sistema funcionara correctamente, logrando la satisfacción del cliente con el producto final.

Capítulo 4: Estudio de la factibilidad

Capítulo 4

4.1 Introducción

Desde los primeros momentos del desarrollo de un *software*, resulta necesario determinar si resultará factible o no. Para ello, se debe realizar un estudio detallado de los beneficios que este aporta y de las inversiones que implicará tanto en la esfera organizativa (entiéndase estructuras, procesos y personas), como en la económica y técnica (teniendo en cuenta habilidades, experiencias, recursos), para llevar a cabo su implementación. Este estudio, incluye una planificación del trabajo a realizar referente al tiempo que demorará el desarrollo del *software*, a partir de la cantidad de personas requeridas y del tamaño del mismo. Una de las tareas de mayor importancia en la planificación de proyectos de *software* es el cálculo de la estimación, que consiste en determinar con cierto grado de certeza, los recursos de *hardware* y *software*, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos. En este capítulo se realiza un estudio de la factibilidad para la realización del sistema propuesto, haciendo una estimación del esfuerzo necesario para llevar a cabo el mismo basado en el método COCOMO II.

4.2 Características del Proyecto

El primer paso a llevar a cabo para la estimación del proyecto consiste en la obtención de los Puntos de Función Desajustados (UFP).

Para calcular los UFP, se deben identificar los siguientes tipos de ítems:

Entradas Externas (*Inputs*)

Salidas Externas (*Outputs*)

Archivo Lógicos Internos (Archivos)

Archivos Externos de Interface (Interfaces)

Consultas externas (*Queries*) (31)

4.2.1 Entradas externas

Se definen como un proceso elemental, mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro, es decir, entrada de datos del usuario o de control que ingresan desde el exterior del sistema para agregar y/o cambiar datos a un archivo lógico interno (32)

Capítulo 4: Estudio de la factibilidad

Entradas externas	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o alta)
Autenticar	1	2	baja
Gestionar usuario	1	3	baja
Mostrar permiso	1	1	baja
Gestionar incidencias técnicas	1	3	baja
Gestionar incidencias de seguridad	1	3	baja
Gestionar solución a incidencias técnicas	1	3	baja
Gestionar solución a incidencias de seguridad	1	3	baja

Tabla # 4.1 Entradas Externas

4.2.2 Salidas externas

Se definen como un proceso elemental con componentes de entrada y de salida, mediante el cual datos simples y datos derivados cruzan la frontera del sistema desde adentro hacia afuera, también se define como salida de datos de usuario o de control que deja el límite del sistema de *software*. (32)

Salidas externas	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o alta)
Generar listado de reportes	1	1	baja
Total		1	

Tabla # 4.2 Salidas Externas

4.2.3 Consultas Externas

Se definen como un proceso elemental con componentes de entrada y de salida, donde un actor del sistema rescata datos de uno o más Archivos Lógicos Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo (lógico interno o de interfaz externo) y los datos de salida no contienen datos derivados. (32)

Capítulo 4: Estudio de la factibilidad

Nombre de la petición	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o alta)
Total		0	

Tabla # 4.3 Consultas Externas

4.2.4 Archivos Lógicos Internos

Constituyen un grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de entradas externas (32)

Nombre del fichero Interno	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o alta)
Incidencias técnicas	1	5	baja
Incidencias seguridad	1	4	baja
Soluciones Incidencias técnicas	1	4	baja
Soluciones Incidencias seguridad	1	3	baja
Total		14	

Tabla # 4.4 Archivos Lógicos Internos

4.2.5 Archivos de Interfaz Externos

Son un grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones, es decir, cada Archivo de Interfaz Externo es un Archivo Lógico Interno de otra aplicación. (32)

Nombre de la entrada externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (baja, media o alta)
Total		0	

Tabla # 4.5 Archivos de Interfaz Externos

La Tabla 4.6 muestra las ponderaciones asociadas a cada tipo de ítem. Estas ponderaciones han sido derivadas y validadas empíricamente mediante la observación de una gran variedad de proyectos.

Capítulo 4: Estudio de la factibilidad

Elementos	Baja	Media	Alta
	Peso	Peso	Peso
Entradas externas	3	4	6
Salidas externas	4	5	7
Consultas externas	3	4	6
Archivos Lógicos Internos	7	10	15
Archivos de Interfaz Externos	5	7	10

Tabla # 4.6 Peso del Factor de Complejidad.

4.3 Estimación inicial

Puntos de Función sin ajustar (UFP): se obtiene a través de la suma del aporte de todos los elementos.

Elementos	Baja		Media		Alta		Aportes
	Cant	Valor	Cant	Valor	Cant	Valor	
Entradas externas	7	3	0	4	0	6	21
Salidas externas	1	4	0	5	0	7	4
Consultas externas	0	3	0	4	0	6	0
Archivos Lógicos Internos	4	7	0	10	0	15	28
Archivos de Interfaz Externos	0	5	0	7	0	10	0
Total							53

Tabla # 4.7 Puntos de Función desajustados

UFP=53.

Una vez que se han obtenido los Puntos de Función sin ajustar del sistema, se puede estimar el esfuerzo, para esto se utilizará el método COCOMO II.

El método COCOMO II consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar estimados para un proyecto. Estas ecuaciones se encuentran ponderadas por ciertos factores de costo, que se influyen en el esfuerzo requerido para el desarrollo del *software*. La meta es obtener un número que caracterice completamente al sistema.

Capítulo 4: Estudio de la factibilidad

4.4 Cálculo del esfuerzo nominal

$$PM_{\text{nominal}} = A * (\text{Size})^E$$

PM nominal: es el esfuerzo nominal requerido en mes-hombre.

A: Es una constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del *software*. El modelo la calibra con un valor de 2.94.

Size: Es el tamaño estimado del *software*, en Puntos de Función sin ajustar (convertibles a KSLOC). Se calcula el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Se utiliza PHP* (factor de conversión = 53 SLOC/UFP).

$$\text{Size} = 53 * 53 = 2809 \text{ SLOC}$$

$$\text{Size} = 2.81 \text{ KSLOC}$$

E: Es una constante, denominada *Factor Escalar*. Se calcula ponderando las variables escalares, mediante la ecuación:

$$E = 0.91 + 0.01 * \sum (Wi)$$

Wi = valor de la variable escalar

Donde las Wi se muestran en la siguiente tabla:

Capítulo 4: Estudio de la factibilidad

Variable	Descripción	Valor
PREC	El sistema es muy familiar	2.25
FLEX	Algo de relajación en cuanto a la flexibilidad del desarrollo	2.50
RESL	La arquitectura es sólida y los riesgos generalmente se mitigan	3.50
TEAM	La interacción del equipo es altamente cooperativa	2.80
PMAT	La madurez del proceso de <i>software</i> es baja	1.80
Total		12.85

Tabla # 4.8 Factor Escalar (Wi)

$$E = 0.91 + 0.01 * 12.85 = 1.167$$

Por tanto:

$$PM \text{ nominal} = A * (\text{Size})^E = 2.94 * (2.81 \text{ KSLOC})^{1.167} = 9.817 \text{ mes-hombre}$$

4.5 Cálculo del esfuerzo ajustado

$$PM \text{ ajustado} = PM \text{ nominal} * \prod (ME_i)$$

Capítulo 4: Estudio de la factibilidad

Multiplicador	Descripción	Valor
PERS	Se tienen analistas y programadores con alta eficiencia y capacidad de trabajo en equipo.	0.90
RCPX	Las exigencias de confiabilidad, documentación y volumen de datos son moderadas, y la complejidad del producto es baja.	1.05
RUSE	No se pretende reutilizar nada.	1.07
PDIF	No existen restricciones en cuanto al tiempo de CPU o al consumo de memoria, la plataforma es muy estable.	1.00
PREX	Tanto los analistas como los programadores tienen aproximadamente 6 meses de experiencia en la aplicación, la plataforma, el lenguaje y las herramientas utilizadas.	0.95
SCED	Se requiere terminar el proyecto en el tiempo estimado.	1.00
FCIL	Se tienen herramientas CASE* simples e infraestructura de comunicaciones básicas.	0.95
Total		0.96

Tabla # 4.9 Multiplicadores de esfuerzo (EM)

$$\Pi (ME_i) = 0.96$$

$$PM \text{ ajustado} = 9.82 * 0.96 = 9.43 \text{ Mes-hombre.}$$

4.6 Cálculo del tiempo de desarrollo, cantidad de hombres y costo

Valores calibrados:

$$A = 2.94; B = 0.91; C = 3.67; D = 0.24$$

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

$$= 0.24 + 0.2 * (1.167 - 0.91) = 0.29$$

Capítulo 4: Estudio de la factibilidad

TDEV (Tiempo de desarrollo) = $C * (PM \text{ ajustado}) ^ F = 3.67 * (9.43) ^ 0.29 = 7.035$ meses

CH (Cantidad de Hombres) = $PM \text{ ajustado} / TDVE = 9.43 / 7.035 \approx 1.34 \approx 1$ persona

Como la cantidad real de hombres disponibles para el desarrollo de la aplicación es 2, al reajustar el tiempo de desarrollo según la cantidad de hombres, resultó un tiempo equivalente a 3.52 meses.

Salario promedio: Para determinar el salario promedio se tuvo en cuenta que los desarrolladores son estudiantes de 5to año, por lo que se toma como salario correspondiente \$100.00

Costo = $2 * 100 * 9.43 = \$1886$

Cálculo de:	Valor
Esfuerzo	9.43 Mes-hombre
Tiempo de desarrollo	3.5 Meses
Cantidad de hombres	2 Personas
Salario medio	\$100.00
Costo	\$1886.00

Tabla # 4.10 Resultados alcanzados

4.7 Beneficios tangibles e intangibles

4.7.1 Tangibles

Teniendo en cuenta que la aplicación no es un producto desarrollado para la comercialización, no es válido mencionar beneficios económicos. Se puede decir que el costo por desarrollar la aplicación es de \$1886.0 MN (moneda nacional), el cual es perfectamente reparable si en un futuro se decide comercializar.

4.7.2 Intangibles

Con el uso de la aplicación se puede reducir considerablemente el tiempo dedicado al desarrollo de las planificaciones, a la vez que se garantiza que no existan errores por doble utilización de recursos, algo que pasa con frecuencia ya que este proceso se desarrolla manualmente en la entidad. En lugares donde no estén automatizadas estas acciones, es necesario que un grupo de personas se dedique a la captura, teniendo que pasar largas horas en ello e incluso muchas veces en horario de sueño, el *software* que se propone, evita este tipo de desgaste físico, así como las dificultades que puede crear no hacerlo de manera correcta, al proveer a los usuarios de un servicio de captura automática que sigue las planificaciones almacenadas en la base de datos.

Capítulo 4: Estudio de la factibilidad

4.8 Análisis de costo y beneficios

El desarrollo de la aplicación no constituye un gasto considerable, pues todas las herramientas que se han empleado en su desarrollo son libres y de código abierto. El sistema está orientado al trabajador y es de fácil aprendizaje. El producto presenta una interfaz amigable y atractiva al usuario, es fácil de usar y navegar por todas las personas que interactúen con la aplicación. Además, no son necesarios los gastos por concepto de tecnología, pues la institución cuenta con la infraestructura necesaria para la implantación de la aplicación. Como el aporte más valioso se tiene la obtención de una aplicación web que brinda toda la información referente a las incidencias.

4.9 Conclusiones

Al finalizar el capítulo se arribó a las siguientes conclusiones:

- El estudio de factibilidad realizado al sistema, demostró que la solución del *software* es factible, pues reportó beneficios sin incurrir en mayores gastos.
- El costo estimado es reparable si se decide comercializar la aplicación.

Conclusiones

Una vez elaborada la investigación se obtuvo una serie de resultados que dan cumplimiento al objetivo propuesto, ellos son:

- La Aplicación Web favorece la gestión de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual, garantizando la agilización de estos procesos, reducción de los gastos producto del consumo de material de oficina, la disminución de los riesgos por pérdida y deterioro de la información aumentando el nivel de seguridad de la misma y la reducción del tiempo de respuesta.
- El objetivo trazado en el diseño teórico – metodológico para el desarrollo de la investigación fue cumplido, comprobándose la hipótesis como respuesta del problema a resolver que originó esta investigación, contribuyendo directamente a la gestión de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual.
- El empleo de los métodos teóricos y empíricos facilitó conocer el estado del objeto de estudio.
- Las entrevistas realizadas a personas encargadas del control de las incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual, sirvió para evaluar el problema y corroboró la necesidad de perfeccionar el proceso de gestión de información de dichas incidencias.
- Las tareas investigativas facilitaron una adecuada organización de la investigación.
- La aplicación Web fue avalada por el cliente para la gestión de información en los procesos de control y seguimiento de incidencias técnicas y de seguridad en la Dirección de Producción Audiovisual

Recomendaciones

Una vez concluido el desarrollo de la investigación se recomienda:

- Planificar una futura iteración, para agregar nuevas funcionalidades al sistema en caso de que surjan nuevos requisitos, por ejemplo: hacer un reporte que muestre las veces que los usuarios han accedido a la aplicación.
- Mejorar la interfaz visual.
- Realizar auditorías periódicas a la aplicación, para controlar su funcionamiento.

Referencias

1. Proceso de informatización 25 de abril 2010, n.º Disponible en: http://www.cubagob.cu/ingles/des_eco/mic/default.html
2. *Gestión*. publicado el: 10 enero de 2010, última actualización: 10 enero Disponible en: http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm.
3. Datos. 10 enero 2010, n.º Disponible en: <http://deconceptos.com/general/dato>.
4. Datos informáticos. 10 enero 2010, n.º Disponible en: <http://deconceptos.com/general/dato>.
5. Términos técnicos 10 enero 2010, n.º Disponible en: <http://tecnologia.glosario.net/terminos-tecnicos-internet/datos-459.html>.
6. HIRT, F. Y. Del libro: «Introducción a los Negocios en un Mundo Cambiante», Cuarta Edición, de Ferrell O. C. y Hirt Geoffrey. 10 enero 2010, n.º p. 121.
7. INFORMATION MANAGEMENT, Qué es gestión de la información. 11 de enero 2010, n.º Disponible en: <http://informationmanagement.wordpress.com/category/gestion/gestion-de-la-informacion>.
8. Woodman L. Information management in large organizations. En: Information management from strategies to action. London: ASLIB; 1985. p. 95-114. 10 enero 2010, n.º
9. Gestión de datos. 10 enero 2010, n.º Disponible en: <http://www.topbits.com/data-management.html>.
10. Incidencia de seguridad. 10 enero 2010, n.º
11. Procesos. 10 enero 2010, n.º Disponible en: http://web.jet.es/amozarrain/Gestion_procesos.htm
12. XPERTA. 20 de enero 2010, n.º Disponible en: <http://www.xperta.es/es/casosdeexito.asp>.
13. Inventario de aplicativos de las CCAA. 19 de enero 2010, n.º Disponible en: [http://plan.aragob.es/circa.nsf/\(IDPublica\)/296709E9B625BB0DC1257194003A7BA1?OpenDocument](http://plan.aragob.es/circa.nsf/(IDPublica)/296709E9B625BB0DC1257194003A7BA1?OpenDocument).
14. Service Desk 20 de enero 2010, n.º p. 1. Disponible en: <http://www.addlink.es/productos.asp?pid=542>.
15. Lenguaje del lado del servidor o cliente 9 de diciembre 2009, n.º Disponible en: http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
16. Perl 9 de diciembre 2009, n.º Disponible en: <http://es.wikipedia.org/wiki/Perl#Descripci.C3.B3n>.

17. Lenguajes de programación. 9 de diciembre 2009, nº Disponible en: <http://www.3djuegos.com/foros/tema/119929/0/lenguajes-de-programacion>.
18. Sistemas gestores de bases de datos. 15 de diciembre 2009, nº Disponible en: <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
19. PECOS, D. MySQL 11 de enero 2010, nº Disponible en: http://www.netpecos.org/docs/mysql_postgres/x57.html.
20. ---. PostGreSQL 11 de enero 2010, nº Disponible en: http://www.netpecos.org/docs/mysql_postgres/x57.html.
21. Qué es SCRUM. 21 de enero 2010, nº Disponible en: <http://www.proyectosagiles.org/que-es-scrum>.
22. Enterprise Architect. 22 de enero 2010, nº p. 1. Disponible en: <http://www.intelegosoft.com/esp/ea/index.asp>.
23. Framework. 15 de enero 2010, nº Disponible en: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
24. Symfony. 15 de enero 2010, nº Disponible en: <http://www.symfony.es/que-es-symfony/>.
25. LEOPOLDO, C. Zend Framework. 15 de enero 2010, nº Disponible en: <http://techtastico.com/post/zend-framework-una-introduccion/>.
26. CakePHP. 15 de enero 2010, nº Disponible en: <http://techtastico.com/post/cakephp>.
27. Requisitos. 5 febrero 2010, nº Disponible en: <http://eva.uci.cu/mod/resource/view.php?id=22044>.
28. Arquitectura. 25 de abril 2010, nº Disponible en: <http://www.csi.map.es/csi/silice/Global71.html>.
29. FUENTES, I. 11 marzo 2010, nº Disponible en: http://informatica.escuelaedib.com/index.php?option=com_myblog&show=Tarjetas-CRC..html&Itemid=59.
30. MAITE RODRÍGUEZ CORBEA, M. O. P. La Metodología XP Aplicable al Desarrollo del Software educativo en Cuba. 13 de marzo 2010, nº Disponible en: http://bibliodoc.uci.cu/TD/TD_0837_07.pdf.
31. ADRIANA GÓMEZ, M. D. C. L. y SILVINA MIGANI, A. O. 11 abril 2010, nº Disponible en: <http://alarcos.inf-cr.uclm.es/DOC/PGSI/doc/teo/8/cocomo2-apuntes.pdf>.
32. Autónoma de Bucaramanga. Obtención y Validación de Modelos de Estimación de Software Mediante Técnicas de Minería de Datos. 11 abril 2010, nº Disponible en: http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r31_art3_c.pdf.

Bibliografía

1. Introducción a los Negocios en un Mundo Cambiante, Cuarta Edición, de Ferrell O. C. y Hirt Geoffrey, McGraw-Hill Interamericana, 2004, Pág. 121.
2. Velázquez, Mairelys Boeras. Sistema de Gestión de Incidencias para la Universidad de las Ciencias Informáticas. Ciudad de La Habana: s. n, 2006. P. 109.
3. [En línea] [Citado el 10 de diciembre del 2009]
<http://www.webestilo.com/asp/asp00.p>.
4. JACOBSON, I., BOOCH, GRADY Y RUMBAUGH. El proceso unificado de desarrollo de software. p. 1
5. Una explicación de la programación extrema (XP*) [En línea] [Citado el 9 de diciembre del 2009] <http://www.apolosoftware.com/>.
6. MANUAL DE PHP*. [En línea] [Citado él: 5 de diciembre del 2009.]
<http://www.manualdephp.es/>
7. Web oficial de PHP*. [En línea] [Citado él: 15 de diciembre del 2009.]
<http://www.php.net/>
8. Larry Wall y otros. Programming PERL*.
9. Web oficial de Symfony. [En línea] [Citado él 25 de enero del 2010]
<http://www.symfony.es/que-es-symfony/>
10. J Mariano González Romano, J Manuel Cordero Valle. Diseño de páginas web. p. 126

Glosario de términos

ASP (Active Server Pages): Páginas Active Server. Entorno para crear y ejecutar aplicaciones dinámicas e interactivas en la Web. Se puede combinar páginas HTML, secuencias de comandos y componentes ActiveX para crear páginas y aplicaciones Web interactivas.

CASE: Acrónimo de *Computer Aided Software Engineering*.

CSS: La hoja de estilo en cascada es un lenguaje formal usado para definir la presentación de un documento estructurado, escrito en HTML o XML.

HTML (HyperText Markup Language): Lenguaje de Marcado de Hipertexto. Lenguaje en el que se escriben las páginas a las que se accede a través de navegadores WWW. Admite componentes hipertextuales y multimedia.

HU: Historia de Usuario

MVC (Modelo Vista Controlador): Es un patrón de arquitectura de *software* que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Perl (Practical Extraction and Report Language): Es un lenguaje de programación desarrollado por Larry Wall inspirado en otras herramientas de UNIX como son: sed, grep, awk, c-shell.

PHP: Lenguaje procesador de *Scripts* servidor. Libre y de código abierto.

RUP (Rational Unified Process): Es una *metodología robusta* que puede ser adaptado a proyectos de mayor o menor complejidad, aplicable a diferentes esferas y ajustable a las necesidades de cada organización.

SGBD (Sistemas Gestores de Bases de Datos): Son un tipo de *software* que sirven de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

SQL (Structured Query Language): Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Las características del álgebra y del cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.

TCP/IP: Es un conjunto de protocolo de red en la que se basa internet y que permiten la transmisión de datos entre redes de computadoras.

UML (Lenguaje Unificado de Modelado): El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

XP (Extreme Programming): Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de *software*.