

Universidad de las Ciencias Informáticas
FACULTAD 6



**Módulo de planificación de tareas para el Sistema de
Captura y Catalogación de Medias**

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

Autor(es): Aliuska López Pelier

José Armando Sánchez Vega

Tutor: Ing. Yoandri Quintana Rondón

Ciudad de la Habana

Junio de 2012



Podrán morir las personas pero jamás sus ideas

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Aliuska López

Firma del Autor

José Armando Sánchez

Firma del Autor

Yoandri Quintana

Firma del Tutor

DATOS DE CONTACTO

Síntesis del tutor:

Nombre y Apellidos: Yoandri Quintana Rondón

Especialidad: Ingeniero en Ciencias Informáticas

Años de Experiencia: 3

Teléfono Particular: 58180170

Dirección electrónica para correspondencia: yqrondon@uci.cu

AGRADECIMIENTOS

De Aliuska:

Quiero agradecer a toda mi familia y en especial a mi adorada madre por todo el apoyo que me ha brindado en mi vida y en estos 5 años en la universidad, que a pesar de todos los problemas hemos sabido salir adelante y superarlos.

Agradecer también a mis amistades de Guantánamo Deyi, Anita, Yania y a los amigos que he hecho a lo largo de mi carrera que han sido muchos, a Piti, Claudia, José, mi Romeo, y todos los que han compartido conmigo. A mis niños de primer año por obedecer siempre a su “madre”, un beso, los quiero. A Luis Angel, un compañero de estudios al cual le debo mucho el haber llegado hasta acá. A Mary que a pesar de todo hemos sido siempre hermanitas y ha sido mi Nana y a Yunier René, por las tantas veces que le molesté para que me prestara la laptop y me ayudara. A mi beba Lara y su hermanito Yoshi por tan buenos momentos. A Sisley, Lewis, Alberto y Ramón por las molestias ocasionadas y por habernos ayudado en el documento y la aplicación. A las amistades de Luis Ángel con las que disfrutamos tan buenos momentos. A todo el colectivo de profesores y estudiantes del proyecto: Osiel, Ale, Yoe, Zori, Yoa, Jean, Arlen, Adnan, Viñolo, Aníbal por el apoyo brindado. Ya mi compañero de tesis José Armando por todo el tiempo que compartimos en la realización de esta tesis.

De José Armando:

A mi madre y mi padre cuyos consejos me guiaron en mis decisiones de formación vocacional y sin los cuales no hubiese llegado hasta aquí.

A mis abuelas y abuelos, que me hicieron el mejor regalo de todos: mi bella familia.

A mis primos Ernesto, Rudy y Ledian y mis primas Tania y Annia, los cuales quiero tanto y los considero más hermanos que de primos.

A mis tías Martha, Alina y Adelaida las cuales siempre han estado pendiente de mis estudios.

A toda mi familia en general la cual quiero con el corazón.

A Yoandri por ser una persona y tutor excelente.

A mi compañera de tesis Aliuska por todo el tiempo que compartimos juntos en la realización de la tesis.

A todos los compañeros del proyecto y profesores de los que tuve la oportunidad de aprender lo que sé y que me sirvió de base para poder graduarme como ingeniero informático.

A mi novia Ivis por ser una persona súper especial y entenderme en todo momento.

A Yasiel, Misael y Angelito junto a quienes pasé toda mi infancia y adolescencia.

A los amigos que tuve la oportunidad de conocer cuando iniciamos la universidad, a las TROPAS y a los que se sumaron cada año. Hay muchos que quizá no tenga la oportunidad de volver a ver, por eso, a todos ellos además: ¡buena suerte!

DEDICATORIA

De Aliuska:

Dedico este trabajo de diploma a la persona más importante de mi vida, la cual ayudó mucho a que yo cumpliera mis sueños, porque solo pensarla me proporcionó fuerzas para enfrentarme a los retos que se me presentaron durante la carrera y así poder superarlos, mi mamá.

Por el amor que me proveen mis abuelos Julia, Luis y Clara, mis tías Gisela, Elena, Adriana, Ileana y mi tío Juan Luis por todo el apoyo brindado. A mis primos Gisela, Alexis, Claudia, Isys, Isme y Papo por todo su cariño. A mi amada madre Mónica por brindarme todo su esfuerzo y dedicación, por ofrecerme su apoyo incondicional y hacer posible la realización de este sueño, para ella todo mi amor.

Lo dedico también a mis vecinos y amigos que han puesto su granito de arena en mi formación. A mi padrastro Israel por ser un padre para mí, así como al grupo de trabajo del proyecto a los que he "molestadó" tanto y en especial a mi amigo tutor Aníbal que me ayudó y apoyó tanto a pesar de la distancia.

A la Revolución, a Fidel, a la UCI y a todos los que confiaron en mí, profesores y amistades, por haberme dado la oportunidad de estudiar y salir adelante a cada momento

De José Armando:

A mis padres, que les debo cuanto he alcanzado, a toda mi familia por apoyarme siempre en todo momento y a la memoria de abuela Isabel y tío Erín.

RESUMEN

El avance en el desarrollo de la humanidad ha impulsado el perfeccionamiento de la planificación, y ello ha conllevado a un auge de la misma, esto a partir de la necesidad en las personas de cumplir sus metas con la calidad requerida. La planificación es utilizada en todos los aspectos de la vida diaria y aplicada en varios sectores, entre los que se encuentran los que manejan materiales audiovisuales, cuya utilización es necesaria para que el trabajo con dichos materiales sea el esperado.

En el presente trabajo de diploma se propone el desarrollo de una herramienta web que permita informatizar la planificación de las tareas de captura, edición y catalogación de medias. Para dar cumplimiento a la propuesta anterior se definen los estilos y patrones arquitectónicos, así como lenguajes de programación, herramientas CASE y entorno de desarrollo integrado con el fin de garantizar el perfeccionamiento de la aplicación. Se exponen además, un conjunto de elementos que ayudan a definir las funcionalidades que va a tener el sistema, como son los requisitos funcionales y no funcionales del mismo y las descripciones correspondientes a los Casos de Uso identificados para su mejor desarrollo y comprensión.

PALABRAS CLAVE

Audiovisuales, catalogación, captura, edición, planificación.

ABSTRACT

Progress in human development has driven the improvement of planning techniques, leading in consequence to a boom of planning use, due to its need in people to achieve their goals with the required quality. Planning is used in all aspects of daily life and applied in all sectors, especially those who involve the work with audiovisual material, which use is critical to achieve the expected results from such materials. In this diploma document, the development of a web tool that allows to computer the planning of media capture, cataloguing and edition tasks is proposed. To fulfill this requirement, architectural styles and patterns, programming languages, CASE tools and integrated development environments are defined, in order to guarantee the software application perfection. A number of factors that help define the system functionalities are also discussed, such as functional and nonfunctional requirements and the corresponding descriptions for the Use Cases identified necessary for a better development and understanding.

KEY WORDS:

Audiovisuals, cataloguing, capture, edit, planning.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Introducción	4
1.2 Conceptos asociados al dominio del problema.....	4
1.3 Objeto de estudio	5
1.3.1 Descripción general.....	5
1.3.2 Descripción actual del dominio del problema.....	6
1.3.3 Situación problemática	7
1.4 Análisis de otras soluciones	8
1.4.1 Tedral	8
1.5 Conclusiones del capítulo	9
CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO	10
2.1 Introducción	10
2.2 Rational Unified Process (RUP)	10
2.3 Lenguaje Unificado de Modelado (UML)	11
2.4 Visual Paradigm como Herramienta CASE	12
2.5 Lenguajes de Programación.....	12
2.5.1 Lenguaje de Programación del lado del servidor	12
2.5.2 Lenguajes de Programación del lado del cliente.....	13
2.6 Tecnología Ajax	14
2.7 Frameworks de desarrollo.....	15
2.7.1 Symfony	15
2.7.2 Dojo.....	16
2.8 Entorno de Desarrollo Integrado (IDE) NetBeans	16
2.9 Correo electrónico.....	17
2.10 Sistema Gestor de Base de Datos PostgreSQL	17
2.11 Conclusiones del capítulo.....	18
CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA	19
3.1 Introducción	19
3.2 Modelo de Dominio	19
3.3 Modelo de Negocio	19
3.4 Requerimientos Funcionales	20

ÍNDICE

3.5 Requisitos No Funcionales.....	21
3.5.1 Requisitos de Rendimiento y Escalabilidad	21
3.5.2 Requisitos de Disponibilidad.....	22
3.5.3 Requisitos de Soporte y Usabilidad.....	22
3.5.4 Requisitos de Software	22
3.5.5 Requisitos de Hardware	22
3.5.6 Restricciones en el diseño y la implementación	22
3.6 Descripción del Sistema Propuesto.....	22
3.6.1 Descripción de los actores.....	22
3.6.2 Diagramas de Casos de Uso del Sistema	23
3.6.3 Descripciones textuales de los Casos de Uso del Sistema.....	23
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA	33
4.1 Introducción	33
4.2 Patrón arquitectónico	33
4.3 Patrones de diseño	34
4.4 Diagramas de Clases del Diseño	34
4.5 Estándar de codificación	37
4.6 Modelo de implementación.....	38
4.9 Pruebas realizadas al módulo	39
4.9.1 Pruebas de Caja Negra	39
4.9.2 Casos de Prueba.....	40
4.10 Resultados de las pruebas	45
4.11 Conclusiones del capítulo.....	45
CONCLUSIONES GENERALES.....	46
RECOMENDACIONES	47
REFERENCIA BIBLIOGRÁFICA	48
BIBLIOGRAFÍA.....	49
GLOSARIO.....	52

ÍNDICE

ÍNDICE DE FIGURAS

Figura 1 Diagrama de clases del Modelo de Dominio	20
Figura 2 Diagrama de Casos de Uso del Sistema	23
Figura 3 Diagrama de clases del Diseño Caso de Uso "Gestionar Planificación"	35
Figura 4 Diagrama de Clases del Diseño Caso de Uso "Notificar por Correo"	35
Figura 5 Diagrama de Clases del Diseño Caso de Uso "Mostrar Planificación por Fecha"	36
Figura 6 Diagrama de Clases del Diseño Caso de Uso "Filtrar por Recurso"	36
Figura 7 Diagrama de Clases del Diseño Caso de Uso "Cargar Datos"	37
Figura 8 Diagrama de Despliegue.....	39
Figura 9 Diagrama de Componentes	39

ÍNDICE

ÍNDICE DE TABLAS

Tabla 1 Descripción de los actores del Sistema	23
Tabla 2 Descripción textual del caso de Uso "Cargar Datos"	24
Tabla 3 Descripción textual del Caso de Uso "Gestionar Planificación"	29
Tabla 4 Descripción textual del Caso e Uso "Buscar tareas Planificadas"	31
Tabla 5 Descripción textual del Caso de Uso "Notificar por Correo"	32
Tabla 6 Caso de Prueba para el Caso de Uso "Cargar Datos"	41
Tabla 7 Caso de Prueba para el Caso de Uso "Gestionar Planificación"	44
Tabla 8 Caso de Prueba para el Caso de Uso "Buscar Tareas Planificadas"	44
Tabla 9 Caso de Prueba para el Caso de Uso "Notificar por Correo"	45

INTRODUCCIÓN

Desde que el hombre comenzó a razonar se ha planteado metas a seguir para cumplir sus objetivos, por ende se ha visto en la necesidad de planificarse. El proceso de planificación es utilizado para realizar planes y proyectos de cualquier tema. Incluso en el desenlace de las grandes batallas de la historia, la victoria siempre ha estado del lado de los grandes planificadores, evidenciando la importancia de una excelente planificación para llevar a buen término cualquier tarea a realizar.

La propia evolución humana y el surgimiento de formas de organización más especializadas, en las que intervienen un gran número de personas y que están presentes en todos los ámbitos de la vida como son: industrias, gobiernos, sistemas de enseñanza, sistemas de salud, entre otros; ha requerido que se realice una mejor planificación y control de las actividades que se ejecutan para lograr propósitos específicos. La evolución de las tecnologías, al mismo tiempo, ha permitido asistir el proceso de planificación mediante la creación de sistemas informáticos capaces de propagar y controlar la información relativa a la planificación de recursos y actividades en diversas instituciones.

Tener una constante planificación es siempre fructífero, un medio donde es vital es en la televisión, donde se pone de manifiesto las actividades que intervienen en el proceso productivo de materiales audiovisuales, como son la catalogación, edición de medias y hasta la transmisión al aire de señales. También permite apoyar otros procesos de interés como es la captura de emisiones de empresas homólogas.

El hecho de planificar deriva una serie de ventajas:

- ✓ Contribuye a actividades ordenadas y con un propósito. Todos los esfuerzos están apuntados hacia los resultados deseados y se logra una secuencia efectiva y sincronizada de tales esfuerzos.
- ✓ Señala la necesidad de cambios futuros. Dirige la atención hacia los objetivos. Ayuda a determinar funcionalidades necesarias.
- ✓ Proporciona una base para el control. La planificación y el control son inseparables, ya que son como los gemelos de la administración. Cualquier intento de controlar sin planes carece de sentido, puesto que no hay forma de que las personas sepan si van en la dirección correcta, a no ser que primero tengan en claro a donde ir. Así, los planes proporcionan los estándares de control.
- ✓ Ayuda a tener siempre presente, por parte de todos los componentes de la organización, los objetivos de esta y la adecuación de ellos al medio, cuando es necesario. (Ramos, 2008)

En Cuba se encuentra la Universidad de las Ciencias Informáticas (UCI), en la cual existen soluciones, tales como el Alfresco para gestión documental, el sitio del horario para la docencia (HorrPublisher), pero ninguno de ellos permite realizar una planificación detallada para mejorar el trabajo realizado con los materiales audiovisuales. En dicha universidad, en el departamento Señales Digitales que

INTRODUCCIÓN

pertenece al Centro de Geoinformática y Señales Digitales (GEYSED), existen proyectos tales como Sistema de Captura y Catalogación de Medias (SCCM) que realizan tareas relacionadas con las medias, de las cuales varias de ellas requieren de una planificación para obtener la calidad necesaria en los resultados. El principal objetivo del SCCM es informatizar los procesos relativos a la gestión de medias y la planificación de las tareas de captura, edición y catalogación de las mismas; la cual, actualmente, se realiza de forma manual y resulta complejo, ya que es necesario hacer un estudio de los canales de transmisión, la frecuencia y horario en que se transmiten los programas; además se requiere tener en cuenta el horario de disponibilidad de los editores y catalogadores. Los estudios mencionados pueden ser muy variables, lo que implica que la planificación pueda ser para corto o largo plazo.

Partiendo de lo descrito anteriormente se ha definido como **problema a resolver**: ¿Cómo garantizar el control de las tareas de captura, edición y catalogación de medias en el Sistema de Captura y Catalogación de Medias (SCCM)? Y para ello se ha tomado como **objeto de estudio** los procesos de planificación de tareas de captura, edición y catalogación. Centrándose esta investigación en el **campo de acción**: la informatización de los procesos de planificación de las tareas de captura, edición y catalogación en SCCM.

El **objetivo general** de la presente investigación es desarrollar un módulo que permita la planificación de tareas de captura, edición y catalogación de medias en SCCM.

Para ello se **defiende la idea** de que el desarrollo de un módulo de planificación de tareas garantizará el control de los procesos de captura, edición y catalogación de medias de manera organizada en SCCM.

Para dar respuesta al objetivo propuesto se trazaron las tareas que a continuación se mencionan:

- ✓ Caracterizar el proceso de planificación de tareas.
- ✓ Caracterizar a nivel nacional e internacional otros sistemas que soportan el proceso de planificación.
- ✓ Seleccionar elementos visuales, conceptuales y técnicos que pueden ser incorporados a la solución propuesta a partir de las caracterizaciones anteriores.
- ✓ Seleccionar las herramientas y procedimientos más factibles para el desarrollo del componente.
- ✓ Realizar el levantamiento de requisitos funcionales y no funcionales.
- ✓ Realizar el diseño para el proceso de planificación de tareas.
- ✓ Implementar un módulo que posibilite la planificación para el Proyecto SCCM.
- ✓ Realizar pruebas de caja negra al módulo de planificación.

INTRODUCCIÓN

Para dar cumplimiento a las tareas planteadas anteriormente, se utilizarán diferentes **métodos científicos**. Entre los **métodos teóricos** a utilizar están:

Análisis y Síntesis: El método permitió recopilar toda la información relativa a soluciones similares existentes, posibilitando arribar a definiciones concretas para el desarrollo de la aplicación de planificación.

Histórico y Lógico: El método permitió, a partir de un estudio realizado previamente al desarrollo del sistema, guiar la idea a defender propuesta para generar los resultados que se obtuvieron con las pruebas realizadas al sistema. Se enfatizó en recopilar información sobre los planificadores de tareas existentes y ver si concordaban con los requisitos necesarios para el sistema propuesto.

Modelación: Mediante la modelación se comprende cómo debe conformarse el módulo, es decir, cómo fue implementado, evidenciándose así el método.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se ofrece información sobre los procesos de planificación de tareas en el mundo: sus características y tendencias, así como la argumentación de sus productos más reconocidos. Se explicarán con más profundidad aspectos relacionados con la situación problemática presentada.

1.2 Conceptos asociados al dominio del problema

La palabra **tarea** reconoce un origen árabe. Proviene del vocablo “tariha” y significa trabajo u obra. Consiste en un hacer que demanda esfuerzo y que generalmente tiene un tiempo límite para su realización. (Real academia española).

Proceso: conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial. (Real academia española).

Un **proceso** según la norma ISO 9000:2005 es un “Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados” (Cruces, 2011).

Según lo planteado anteriormente no se puede confundir el término tarea con el término proceso, porque tarea se emplea para designar a aquella acción que generalmente demanda de parte de quien la realiza cierto esfuerzo y presenta un tiempo limitado, es decir, un tiempo límite para su realización y proceso es la realización de varias acciones que se relacionan mutuamente, en la cual los elementos de entrada se convierten en resultados.

La **planificación** es el proceso metódico diseñado para obtener un objetivo determinado. (Real academia española). En términos generales, por **planificación** se refiere a la acción o resultado de planificar alguna cuestión, tarea o actividad que lleva a cabo un ser humano. Después de lo citado anteriormente la **planificación** debe ser entendida básicamente como un proceso que implicará la observación de una serie de pasos que se establecerán a priori.

Captura no es más que la acción y efecto de capturar. (Real academia española). En esta investigación la palabra **captura** se utiliza para definir toda aquella operación o conjunto de operaciones que se realizan con el fin de obtener datos provenientes de señales digitales para transferir hacia un soporte de almacenamiento. A este conjunto de operaciones se le denomina proceso de captura.

Catalogación: según la Real Academia de la Lengua Española (RAE), es la acción de apuntar, registrar ordenadamente o clasificar un objeto según normas de ordenación, formando catálogo con ellos. Se puede asumir que la **catalogación** es todo el procedimiento que se realiza para almacenar ordenadamente y clasificar un conjunto de contenidos, en este caso audiovisuales.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Edición: presentar en pantalla los resultados de una operación realizada con el ordenador. (EIPaís, 2011)

En términos informáticos edición, se puede asumir como la acción de abrir un documento en una computadora con la posibilidad de modificarlo, refiriéndose en esta investigación a la edición de materiales audiovisuales.

1.3 Objeto de estudio

A continuación se describen los procesos que constituyen el objeto de estudio de la investigación: la planificación de las tareas de captura, edición y catalogación de medias. Se exponen los conceptos y generalidades asociados a estos procesos.

1.3.1 Descripción general

1.3.1.1 Planificación

En informática la planificación es el proceso sistemático diseñado para obtener un objetivo determinado, donde esta puede ser a corto o a largo plazo, y establece vías que conducen al cumplimiento de las metas previamente trazadas. Esta se realiza de dos formas, manual o mediante un programa informático que resuelva tales especificaciones; manual es algo tedioso porque hay que estar horas pensando cómo se va a planificar y escribiendo, y mediante un software es más cómodo ya que solo es especificarle los campos requeridos y el mismo notifica los datos de la tarea mediante un correo u otra vía. En el proyecto SCCM no existe ninguna aplicación capacitada para planificar tareas, dado que se realiza de forma manual. La misma se registra en un documento Excel, archivando las actividades de la semana y el responsable de cada una. Posterior a esto, se le da a conocer a cada miembro las tareas asignadas.

1.3.1.2 Captura de medias

En informática la transferencia de cualquier tipo de datos al disco duro de un ordenador se le conoce como captura, específicamente la captura de medias no es más que la grabación de sonido, audio o video (Pérez, 2009). En el proyecto SCCM no existe ningún módulo que resuelva este problema, pero se ha conceptualizado la captura de video mediante la interacción con una tarjeta que cumpla con los requisitos necesarios para realizar dicha operación.

1.3.1.3 Catalogación de medias

La catalogación de medias es la inclusión y edición de características y datos descriptivos de los materiales, que se almacenan junto a ellos para facilitar los procesos de búsqueda y recuperación. En

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

SCCM existe un módulo que realiza la catalogación, es decir, permite describir los materiales audiovisuales, el mismo es una aplicación web y multiplataforma. Además, incluye la ejecución de una serie de actividades como son: la búsqueda o recuperación de los archivos audiovisuales digitalizados, la visualización del material y la descripción del mismo por parte del documentalista. Este es uno de los procesos fundamentales en el proyecto, ya que los materiales son enriquecidos con una mayor cantidad de información relacionada con su contenido en aras de lograr un proceso de búsqueda precisa.

1.3.1.4 Edición de medias

En informática, la edición de medias es un proceso mediante el cual se puede mejorar y corregir los errores de una grabación. Para ello se necesita reproducir la fuente, realizar un cortado de la misma para así seleccionar los fragmentos de vídeo o audio, permitiendo agregarle efectos, filtros, textos y transiciones que formarán parte del resultado final.

1.3.1.5 La planificación de tareas en SCCM

La planificación de tareas en SCCM permite organizar la recepción, ordenamiento, clasificación y modificación de los materiales audiovisuales, teniendo en cuenta que puedan ser para corto o largo plazo. Todas estas requieren del nombre, día, tiempo de duración y descripción, además de tener un recurso y atributo específico. Dicho proceso ayuda a la organización y cumplimiento de las tareas que se realicen.

- ✓ En el caso de una tarea de captura, los recursos son las tarjetas y el atributo específico, los canales.
- ✓ Para las tareas de catalogación, los recursos son los usuarios documentalistas y el atributo específico, los materiales disponibles a catalogar.
- ✓ Para las tareas de edición, los recursos son los usuarios editores y el atributo específico, los materiales a editar.

1.3.2 Descripción actual del dominio del problema

La Universidad de las Ciencias Informáticas (UCI), es una institución que tiene entre sus objetivos la formación integral y destaca también el ser líder en la producción, apoyándose para esto en el trabajo que se realizan en los centros de desarrollo de cada facultad, priorizando desde sus inicios la producción del software necesario para brindar servicios tanto a nivel institucional como nacional e internacional. A la Facultad 6, perteneciente al centro de desarrollo GEySED, se le ha designado la tarea de construir un sistema distribuido que constituya una solución integral para la captura, edición y

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

catalogación de medias, desarrollando un grupo de sistemas que juntamente con el módulo de planificación de tareas conformarían integralmente la solución antes mencionada.

El departamento de Señales Digitales es el principal proveedor de software y servicios de la Dirección de Televisión Universitaria (DTU) en la UCI. La DTU tiene entre sus principales funciones, la expansión de materiales audiovisuales en la universidad. Se identificaron la DTU y el ICRT¹, porque actualmente en dichas instituciones no existe ningún software que brinde las prestaciones de catalogación, publicación y gestión de medias, a esto se le añade que la planificación de tareas se realiza de forma manual, por lo que se necesita de un sistema que automatice este proceso para mejorar en tiempo y realizarlo de una forma más simple. Es importante resaltar que existen actualmente otras soluciones que brindan servicios similares a los que se pretenden desarrollar, las cuales son de carácter propietario y presentan un costo muy elevado. La universidad debe potenciar la economía del país mediante la producción de software, por lo que actualmente desarrolla aplicaciones para empresas nacionales e internacionales. Además, debe abastecerse internamente de todo el software necesario, el que en algún momento puede llegar a ser comercializable. Esto constituye una de las razones para el desarrollo de este producto, que es un módulo fundamental de la solución descrita con anterioridad. El sistema debe permitir que el planificador pueda añadir, modificar, eliminar tareas y revisar cuales son las actividades que se tienen concebidas en el día, semana y en el mes, garantizando que se planee un recurso específico. Además de notificar por correo, buscar las tareas según un criterio, que en este caso serían los recursos.

1.3.3 Situación problemática

La planificación es importante en todos los aspectos de la vida diaria, pero cuando se realiza manual es un proceso complejo, porque se necesita hacer un estudio detallado de los procesos y recursos que intervienen en ello y estos análisis pueden ser muy variables. En cualquier institución que trabaje con materiales audiovisuales y la planificación se realice de forma manual, puede incurrir en diferentes errores, debido a que las complejidades de las tareas de captura, edición y catalogación de medias suelen ser elevadas. Ejemplo de esto, es cuando se realiza un proceso de captura donde debe planificarse con anterioridad aquellas medias que serán grabadas y por ello las personas encargadas del proceso de grabación deben estar pendientes en todo momento para iniciar dicha acción.

La informática se ha convertido en una herramienta indispensable para facilitar este proceso. En la universidad, en el proyecto SCCM, se realiza la planificación de las tareas de captura, edición y catalogación de medias de forma manual. En el mismo se hace necesario realizar un estudio de los

¹ Instituto Cubano de Radio y Televisión

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

canales de transmisión, la frecuencia y horario en que se transmiten los programas, que en varias ocasiones ocurren errores, tales como que se le asigne una misma tarjeta a varios canales en el mismo horario. Además, este mismo proceso puede ocurrir para los usuarios de catalogación y edición, donde se puede cometer el error de asignarle dos materiales al mismo tiempo, lo que provocaría que se dejaran de realizar algunas actividades por mala planificación. A partir de lo descrito anteriormente se identificó la necesidad de realizar una aplicación que permita controlar el proceso de planificación para mejorar el trabajo que se realiza con materiales audiovisuales.

1.4 Análisis de otras soluciones

Existen compañías que han puesto empeño en el desarrollo de sistemas de gestión de medias que posibiliten la planificación de sus procesos. En este subíndice se presenta una serie de productos que tienen que ver de una forma u otra con lo que se quiere realizar en el proyecto SCCM.

1.4.1 Tedral

Tedral es una empresa dedicada a la elaboración de software para el procesamiento de media, ya sean sistemas de vigilancia, de gestión y almacenamiento o aplicaciones para visionado de materiales. En esencia, la empresa presenta una serie de aplicaciones que de una forma u otra permiten adquirir experiencias para el desarrollo de sistemas semejantes. Especialmente para su gestión y almacenamiento construye el módulo de indexación y catalogación automática TD Indexer, el cual contiene sub-módulos de captura e indexación automática. (Pérez, 2009)

Dentro de las aplicaciones de Tedral se encuentra Ingest Control (IC) que tiene como objetivo la automatización y gestión de entradas procedentes de diversas fuentes, en la cual se encuentra la interfaz de usuario IC Planning que permite configurar las planificaciones realizadas sobre los distintos recursos disponibles. Sus características y acciones son:

Características:

- ✓ Configurar los distintos recursos.
- ✓ Asignar la planificación a los recursos.
- ✓ Monitorizar la consola de mensajes de sistema.

Acciones:

- ✓ Configurar Centros, Recursos y Canales.
- ✓ Definir reglas de planificación.
- ✓ Definir los flujos asociados a las distintas reglas.
- ✓ Importar recursos desde XML generado por aplicación de testeo de capturas.
- ✓ Importar/Exportar la información completa de un Centro.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- ✓ Monitorizar la consola de mensajes.
- ✓ Consultar el estado de ejecución de la planificación y del agente controlador.

Se identificó TEDIAL con la interfaz IC Planning como una solución, pero no se puede utilizar porque es propietaria y la misma presenta elevados costos en el mercado. Se verificaron, otras soluciones tales como MediaBox, VideOma, VideoIngest, las cuales no permiten acceso a la información necesaria para asegurar la existencia de funcionalidades que garanticen la planificación de sus procesos, porque son propietarios y no se tiene acceso a la documentación detallada de cada uno de ellos.

1.5 Conclusiones del capítulo

En este capítulo se realizó un estudio de los principales conceptos asociados a los términos y procesos asociados a la planificación, lo que permitió un mejor entendimiento del tema a tratar. El estudio de otras soluciones permitió justificar la necesidad de realizar un módulo de planificación con herramientas libres, porque la única con una propuesta similar (IC Planning) va en contra de la soberanía tecnológica por la que aboga el país.

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO

2.1 Introducción

En el presente capítulo se realiza un estudio sobre las herramientas y tecnologías utilizadas para el desarrollo de la aplicación, así como la metodología de desarrollo, los lenguajes de programación y el gestor de base de datos.

2.2 Rational Unified Process (RUP)

En español Proceso Unificado de Desarrollo (RUP), es una metodología que se utiliza para la Ingeniería de Software, va más allá del análisis y el diseño orientado a objetos, para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. (Jacobson, 2000).

RUP mediante los casos de uso define las funcionalidades, y las crea a través de la arquitectura, lo que sin lugar a dudas propicia una interacción necesaria entre casos de uso y arquitectura. Los principales elementos de esta metodología son los trabajadores, sus actividades, los artefactos y el flujo de actividades que es el que muestra el resultado observable.

Beneficios que aporta RUP:

- ✓ Permite desarrollar aplicaciones sacando el máximo provecho de las nuevas tecnologías, mejorando la calidad, el rendimiento, la reutilización, la seguridad y el mantenimiento del software mediante una gestión sistemática de los riesgos.
- ✓ Permite que la producción cumpla con las necesidades de los usuarios, a través de la especificación de los requisitos, con una agenda y costo predecible.
- ✓ Enriquece la productividad en equipo y proporciona prácticas óptimas de software a todos sus miembros.
- ✓ Permite llevar a cabo el proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades críticas.
- ✓ Unifica todo el equipo de desarrollo de software y mejora la comunicación al brindar a cada miembro del mismo una base de conocimientos, un lenguaje de modelado y un punto de vista de cómo desarrollar software.
- ✓ Optimiza la productividad de cada miembro del equipo al poner al alcance la experiencia derivada de miles de proyectos y muchos líderes de la industria. (Jacobson, 2000)

Teniendo en cuenta lo antes planteado, se escoge RUP, ya que unifica los mejores elementos de otras metodologías, lo que garantiza que sea de primer nivel; una de las razones fundamentales es que está

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO

preparada para desarrollar proyectos grandes y complejos, además de estar definida por tres aspectos esenciales:

- ✓ Dirigido por casos de uso: Basándose en los casos de uso, los desarrolladores crean una serie de modelos de diseño e implementación que los llevan a cabo. Además, estos modelos se validan para que sean conformes a los casos de uso. Finalmente, los casos de uso también sirven para realizar las pruebas sobre los componentes desarrollados.
- ✓ Centrado en la arquitectura: En la arquitectura de la construcción, antes de construir un edificio este se contempla desde varios puntos de vista: estructura, conducciones eléctricas, fontanería, etc. Cada uno de estos aspectos está representado por un gráfico con su notación correspondiente. Siguiendo este ejemplo, el concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema.
- ✓ Iterativo e incremental: Todo sistema informático complejo supone un gran esfuerzo que puede durar desde varios meses hasta años. Por lo tanto, lo más práctico es dividir un proyecto en varias fases. Actualmente se suele hablar de ciclos de vida en los que se realizan varios recorridos por todas las fases.

Por otra parte, también es imprescindible para la solución del sistema porque genera una documentación muy amplia, que es de gran importancia para el desarrollo, ya que un desarrollador puede pedir la baja y solo leyendo la documentación que se generó se puede seguir adelante con la continuidad del software. Además, el módulo que se propone realizar forma parte de un sistema que contiene varios módulos, integrándose y formando una aplicación ideal para aplicar dicha metodología.

2.3 Lenguaje Unificado de Modelado (UML)

UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema que involucra una gran cantidad de software; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Cada diagrama tiene fines distintos dentro del proceso de desarrollo. UML no es una guía para realizar el análisis y diseño orientado a objetos, es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos (Orallo, 2009).

La utilización del UML posee gran ventaja ya que permite el entendimiento y asimilación por parte del equipo de trabajo de una mejor manera. Además, posibilita la detección y resolución de errores de forma más ágil y la documentación del software es confeccionada con mayor rapidez y organización.

Se ha seleccionado UML para la modelación del sistema, a partir de que:

La metodología seleccionada utiliza UML como lenguaje de modelado. Además, consta de todos los elementos y diagramas que permiten modelar los sistemas sobre la base del paradigma orientado a objetos. Los modelos orientados a objetos cuando se construyen en forma correcta, son fáciles de

comunicar, cambiar, expandir, validar y verificar. Este modelamiento en UML es flexible al cambio y permite crear componentes plenamente reutilizables. (Orallo, 2009).

En este caso específico se desea realizar un sistema orientado a objeto y este lenguaje unificado es el más apropiado.

2.4 Visual Paradigm como Herramienta CASE

Las Herramientas CASE (Computer-Aided Software Engineering: Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. (Together, 2010)

Se escoge esta herramienta para la modelación del sistema porque ofrece muchas ventajas por sus características, como ser multiplataforma, ser gratis (en su edición Community) y soportar el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Genera códigos, exporta e importa diagramas en el estándar XML y como imágenes. Además, permite el modelado de bases de datos y de diagramas de clases.

2.5 Lenguajes de Programación

Un lenguaje de programación es un lenguaje artificial usado para controlar el procedimiento de una máquina, principalmente una computadora. Se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas por máquinas.

2.5.1 Lenguaje de Programación del lado del servidor

Como refiere el título del epígrafe, este lenguaje es el que se ejecuta en el servidor web. El mismo construye las páginas web antes de enviarlas al navegador, atendiendo las peticiones hechas por el cliente. Los códigos escritos en el mismo pueden acceder a bases de datos, recursos en la red y ficheros en el propio servidor. El escogido para desarrollo de la aplicación es el siguiente:

2.5.1.1 PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-Processor (inicialmente PHP Tools, o, Personal Home Page Tools). Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. (López, 2008)

Este lenguaje fue escogido para la creación del sistema de planificación por que permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, etc. También tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como Unix (y de ese tipo, como Linux o Mac OS X) y Microsoft Windows. Además es orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. Permite aplicar técnicas de programación orientada a objetos. Presenta una biblioteca nativa de funciones sumamente amplia e incluida y es totalmente libre, por lo que se presenta como una alternativa de fácil acceso para todos.

2.5.2 Lenguajes de Programación del lado del cliente

Los lenguajes de programación que se ejecutan del lado del cliente son independientes del servidor e interpretados directamente por los navegadores web. Los lenguajes de programación del lado del cliente que se han seleccionado para el desarrollo de la aplicación son los siguientes:

2.5.2.1 HTML

HTML, por sus siglas en inglés Hyper Text Markup Language, es decir, Lenguaje de Marcas de Hipertexto. El HTML permite describir la estructura y el contenido en forma de texto, además de complementar el texto con objetos tales como imágenes. Este lenguaje se escribe mediante etiquetas, que aparecen especificadas por corchetes angulares (< y >). (WordPress, 2010)

Se escoge HTML ya que las páginas web pueden ser vistas por el usuario mediante el navegador. Además facilita el trabajo ya que en un documento HTML se pueden ver tres partes bien diferenciadas:

- ✓ Una cabecera en la que se especifica el tipo de documento HTML.
- ✓ El cuerpo del documento, donde se coloca toda la información que contiene la página Web.
- ✓ Una cabecera de documento, donde se especifica la información del documento.

2.5.2.2 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Las características principales que lo hacen factible para su utilización son:

- ✓ Es simple, no hace falta tener conocimientos avanzados de programación para poder utilizarlo.
- ✓ Maneja objetos dentro de la página Web y sobre ese objeto se puede definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario.
- ✓ Es dinámico, responde a eventos en tiempo real. Tales como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO

un tiempo. Con esto se puede cambiar totalmente el aspecto de la página al gusto del usuario, evitando tener en el servidor una página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario, etc.

- ✓ Se utiliza principalmente del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. (Carrión, 2008)

2.5.2.3 CSS

CSS (Cascading Style Sheets) conocida también como hojas de estilo en cascada es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

Algunas ventajas de utilizar CSS son:

- ✓ Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- ✓ Separación del contenido de la presentación, lo que facilita al creador, diseñador, usuario o dispositivo electrónico que muestre la página, la modificación de la visualización del documento sin alterar el contenido del mismo, sólo modificando algunos parámetros del CSS.
- ✓ Mejora en la accesibilidad del documento, pues con el uso del CSS se evitan antiguas prácticas necesarias para el control del diseño (como las tablas), y que iban en perjuicio de ciertos usos de los documentos, por parte de navegadores orientados a personas con algunas limitaciones sensoriales. (SANTIAGO, 2012)

Estas ventajas hacen que sea necesaria la utilización de CSS, para darle estilo y se vea organizada la aplicación.

2.6 Tecnología Ajax

AJAX es una técnica de desarrollo web para crear aplicaciones interactivas, integra un conjunto de tecnologías obteniendo los mejores méritos de cada una de ellas. Además incorpora una presentación basada en estándares XHTML y CSS; además, exhibe una interacción dinámica utilizando Document Object Model (DOM), para la manipulación y el intercambio de los datos utiliza XML and XSLT; también propone en pos de la recuperación de datos asíncrona XML Http Request; además de la integración a todos sus atributos de JavaScript.

Una aplicación Ajax se basa en introducir un intermediario entre el usuario y el servidor, haciéndola más reactiva y rápida. Su explicación más sencilla es que al inicio de la sesión el navegador carga un Motor Ajax en vez de cargar una página Web como lo hace comúnmente. Este Motor que está escrito

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO

en JavaScript se hace responsable de renderizar la interfaz de usuario y de comunicarse con el servidor en nombre del usuario. (Pérez, 2008)

Esta tecnología será de gran utilidad para realizar las funcionalidades sobre la estructura de datos, ya sea para cargar las tareas, modificarlas, eliminarlas y buscar los datos para lograr el auto completamiento en los campos necesarios, realizando las peticiones al servidor de manera asíncrona y evitando que la interfaz visual sobre la que trabaja el usuario se refresque cada vez que realice una de estas operaciones.

2.7 Frameworks de desarrollo

Los frameworks de desarrollo constituyen una herramienta de extrema importancia para la implementación de aplicaciones en la actualidad. Proporcionan a los programadores y diseñadores una mejor organización y estructura de sus proyectos, lo cual acelera el desarrollo del producto final. La utilización de un framework adecuado proporciona patrones utilizados para resolver tareas comunes, encapsulando operaciones complejas en instrucciones sencillas.

De forma breve, un framework es una estructura conceptual y tecnológica compuesta por bibliotecas, componentes y clases que facilitan el desarrollo ágil, seguro y escalable de una aplicación.

2.7.1 Symfony

Es un framework que 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.

Es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (Zaninotto, 2008)

Las características que hacen indispensables que se utilice Symfony en el sistema son:

- ✓ 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.

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO

- ✓ 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 plugins.
- ✓ Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo. (Eguiluz, 2012)

También es de gran importancia ya que está elaborado sobre el patrón arquitectónico Modelo Vista Controlador que organiza el código en base a su función, separándolo en tres partes: el modelo, donde se define la lógica del negocio, la vista, que permite a los usuario interactuar con el sistema, y el controlador, que realiza las llamadas al modelo para obtener los datos y pasárselos a la vista

2.7.2 Dojo

Dojo es un conjunto de herramientas open source DHTML (Dynamic HTML) escrito en JavaScript que contiene APIs (librerías) y widgets (controles) que permiten facilitar el desarrollo de aplicaciones web sobre tecnología AJAX. Este toolkit (contenedor) funciona como un sistema de empaquetado inteligente que relaciona la abstracción de eventos, almacenamiento de APIs en el cliente, efectos de interfaz de usuario y su interacción con AJAX. (Russell, 2008)

Las funcionalidades de las que dispone Dojo están divididas en tres proyectos: Dojo Core, Dijit y Dojox. Dijit y Dojox están basadas en la sólida base que Dojo Core brinda. Dijit incorpora un conjunto de widgets con los cuales se pueden crear amigables interfaces en muy poco tiempo, escribiendo casi o ningún código JavaScript. Los mismos son factibles para el trabajo con los grid, que son de gran utilidad para el desarrollo del sistema de planificación a la hora de mostrar las tareas, permitiendo consultar los datos de estas visualmente. La utilización de este framework de JavaScript permite agilizar el desarrollo final de la aplicación por las facilidades que brindan sus clases y componentes. Por lo antes planteado se decide escoger Dojo para el desarrollo ya que presenta características que son de gran utilidad, también es de destacar que esta biblioteca es de código abierto.

2.8 Entorno de Desarrollo Integrado (IDE) NetBeans

NetBeans es una de las soluciones más completas para el desarrollo de aplicaciones web con elevado contenido gráfico. Es un IDE de código abierto escrito sobre la base de la tecnología Java, que ofrece todos sus beneficios de forma gratuita. Soporta óptimamente lenguajes como PHP, HTML y CSS y establece una buena integración con el framework Symfony. Además, cuenta con un editor de textos, resalta las sintaxis, permitiendo mejor entendimiento del código y compila en tiempo real.

Se identifica este IDE para el proyecto SCCM en las aplicaciones web, donde se encuentra la de planificación de tareas.

2.9 Correo electrónico

El correo electrónico usa una arquitectura cliente/servidor. Para permitir la transferencia de mensajes existe una variedad de protocolos de red que permiten que diferentes máquinas con diferentes sistemas operativos y usando distintos programas de correo, envíen y reciban email. La aplicación de planificación utilizará el protocolo de transferencia de correo SMTP², que es un protocolo simple de transferencia de correo. Para poder enviar correo, el cliente envía el mensaje a un servidor de correo saliente, el cual contacta con el servidor de correo de destino para la entrega. Por esta razón, es necesario especificar un servidor SMTP cuando se esté configurando un cliente de correo. No requiere autenticación, por ende, cualquiera en la Internet puede enviar correo a cualquier otra persona o grupo de personas. Esta característica de SMTP es lo que hace posible el correo basura o spam. (CAJAS, 2007)

Se utiliza el envío de correo para notificarle al usuario que debe realizar una tarea determinada, donde se le especifica los datos correspondientes a esta. En el caso de las tareas de captura como no se le asigna a un usuario específico, se le notifica al planificador.

2.10 Sistema Gestor de Base de Datos PostgreSQL

PostgreSQL es un Sistema Gestor de Base de Datos que ha sido desarrollado desde 1977. Comenzó como un proyecto en la Universidad Bekerley de California. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. Está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre. (González, 2012)

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. (Martinez, 2010)

La última serie de producción es la 9.1, a pesar de que la aplicación de planificación puede funcionar con versiones anteriores. Sus características técnicas lo hacen uno de los sistemas gestor de bases de datos más potentes y robustos del mercado. Entre sus principales características se puede destacar

² **Simple Mail Transfer Protocol (SMTP)** Protocolo Simple de Transferencia de Correo.

CAPÍTULO 2: HERRAMIENTAS Y TECNOLOGÍAS A UTILIZAR PARA EL DESARROLLO

que funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (Martinez, 2010)

La investigación realizada permite proponer como gestor de base de datos a PostgreSQL, pues sus ventajas y prestaciones son amplias y ajustables al problema presentado.

2.11 Conclusiones del capítulo

Luego de realizar el estudio de sus características, han sido seleccionadas las tecnologías y herramientas a utilizar en la construcción de la solución que se propone de acuerdo a las ventajas que las mismas reportan, las cuales garantizan que el software tenga la calidad y el coste requerido, porque como la aplicación es desarrollada bajo tecnologías libres no tendrían que pagar licencia, además de ahorrar material de oficina. La selección realizada tuvo en cuenta que las tecnologías fueron libres, siendo esta investigación un apoyo a la política de soberanía tecnológica por la que aboga el país.

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

En el presente capítulo se realiza el análisis de la solución propuesta para un mejor entendimiento de las funcionalidades con las que contará el software. Se presentan los diagramas de caso de uso, así como los requisitos funcionales y no funcionales, además de describir los actores que intervienen en el sistema, mostrándose los diagramas y sus descripciones.

3.2 Modelo de Dominio

Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Se crea para documentar los conceptos dominantes y el vocabulario del mismo. Identifica las relaciones entre todas las entidades importantes dentro de él e identifica generalmente sus métodos y cualidades importantes. Se puede utilizar con eficacia para verificar y validar la comprensión del dominio del problema entre los varios poseedores de apuestas del grupo del proyecto. Es especialmente provechoso como una herramienta de la comunicación. (Teleformacion, 2012)

3.3 Modelo de Negocio

Permite obtener una visión de la organización que permita definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos. El objetivo del Modelo de Negocio es describir los procesos, existentes u observados, con el propósito de comprenderlos. Se especifican qué procesos del negocio soportará el sistema. Además de identificar los objetos del dominio o del negocio, implicados. (Telefomación, 2008)

Como el sistema que se va a desarrollar es complejo, no se realizó un modelado completo de los procesos de negocio, por lo que se decidió realizar un modelo de dominio. Su objetivo es comprender y describir las clases conceptuales más significativas dentro del dominio del problema, lo cual permite definir los procesos más relevantes. Esto ayuda a los usuarios, desarrolladores y clientes a tener un vocabulario común, contribuyendo con la aceleración del desarrollo del sistema.

Diagrama de clases del Modelo del Dominio

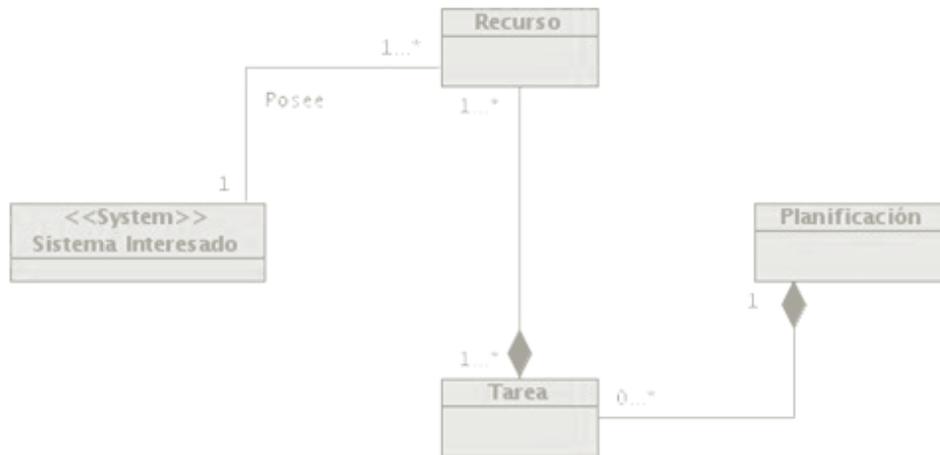


Figura 1 Diagrama de clases del Modelo de Dominio

Descripción del diagrama de clases del dominio

A continuación se muestra una definición de las clases conceptuales del Modelo del Dominio para ayudar a un mejor entendimiento de los procesos identificados. Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades (Larman, 2000). Las clases del Modelo de Dominio son abstracciones de conceptos del mundo real, no representan clases del código de una aplicación, sino las relaciones que existen entre los elementos que le dan origen o intervienen en el proceso que se informatizará.

Descripción de clases

Tarea: Representa una actividad que se realiza en el sistema interesado y que requiere ser planificada.

Planificación: Constituye el resultado final de planificar las tareas y los recursos de un sistema interesado, será la guía a seguir para la ejecución de las actividades en dichos sistemas.

Recurso: Son los elementos del sistema interesado que se deberán asignar en cada planificación de tarea.

Sistema Interesado: Representa a los sistemas que requieren la planificación de tareas y recursos, con los que se establecerá comunicación para realizarles el servicio de planificación.

3.4 Requerimientos Funcionales

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el sistema. (Figuerola, 2007)

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

RF1: Cargar Datos.

Descripción: El sistema debe mostrar la interfaz con los recursos (tarjeta, usuario de edición y usuario de catalogación) existentes en la Base de Datos.

RF2: Crear Tarea de Planificación.

Descripción: El sistema debe permitir crear una tarea de planificación con los datos necesarios para la misma, pudiendo ser nombre, hora inicio, hora fin, descripción de la misma.

RF3: Modificar Tarea Planificación.

Descripción: El sistema debe permitir modificar una tarea de planificación existente.

RF4: Eliminar Tarea Planificación.

Descripción: El sistema debe permitir eliminar una planificación existente.

RF5: Mostrar Planificación por Fecha.

Descripción: El sistema debe mostrar las planificaciones realizadas por día, semana o mes.

RF6: Filtrar por Recurso.

Descripción: El sistema debe mostrar las tareas planificadas de acuerdo al recurso seleccionado.

RF7: Notificar por correo.

Descripción: El sistema debe enviar una notificación por correo al usuario con los datos de la tarea asignada, ya sea cuando se cree, modifique o elimine la misma.

3.5 Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez 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. (Oriente, 2008) O sea, que definen características que hacen un producto confiable y atractivo.

3.5.1 Requisitos de Rendimiento y Escalabilidad

La aplicación se desarrolla de forma tal que pueda ser usada en futuras versiones del sistema. Además debe ser eficiente a las peticiones realizadas en cada momento. La base de datos deberá resistir una alta concurrencia, así como un gran número de operaciones simultáneas sobre la misma.

3.5.2 Requisitos de Disponibilidad

El sistema debe estar disponible las 24 horas, los 7 días de la semana.

3.5.3 Requisitos de Soporte y Usabilidad

La aplicación permite la modificación de la misma para asegurar su extensibilidad y lograr mejores prestaciones. Además fue concebida para ser utilizada por personas que tengan conocimientos básicos de informática, siendo más avanzado en el trabajo con archivos multimedia.

3.5.4 Requisitos de Software

En las estaciones clientes:

Sistema Operativo: Cualquier Sistema Operativo

Navegador: Mozilla Firefox y Google Chrome

En los servidores:

Servidor Web: Apache

Framework: Symfony y Dojo

Sistema Operativo: Linux (Ubuntu)

Sistema Gestor de Base de Datos: PostgreSQL

3.5.5 Requisitos de Hardware

En las estaciones clientes:

Memoria: 256 MB Procesador: Intel Celeron D

En los servidores:

Procesador: Intel(R) Core(TM) 2 Duo

Memoria RAM: 2048 MB

3.5.6 Restricciones en el diseño y la implementación

El sistema será implementado siguiendo el paradigma de la Programación Orientada a Objetos, se utilizarán los frameworks, Symfony y Dojo. El módulo de interacción con el usuario final es una aplicación web y es multiplataforma.

3.6 Descripción del Sistema Propuesto

El módulo de planificación está destinado para ser utilizado por empresas y personas que necesiten un planificador de tareas para mejorar el trabajo con materiales audiovisuales. Se debe configurar en un fichero XML los parámetros de conexión con las Bases de Datos, así como los recursos que se desean planificar de los sistemas interesados. Además permite configurar si se desea o no notificar por correo.

3.6.1 Descripción de los actores

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

Actor	Descripción
Planificador	Es el encargado de inicializar los CU Crear Tarea de Planificación, Modificar Tarea de Planificación, Eliminar Tarea de Planificación, Mostrar Planificación por Fecha, Filtrar por Recurso y Cargar Datos.

Tabla 1 Descripción de los actores del Sistema

3.6.2 Diagramas de Casos de Uso del Sistema

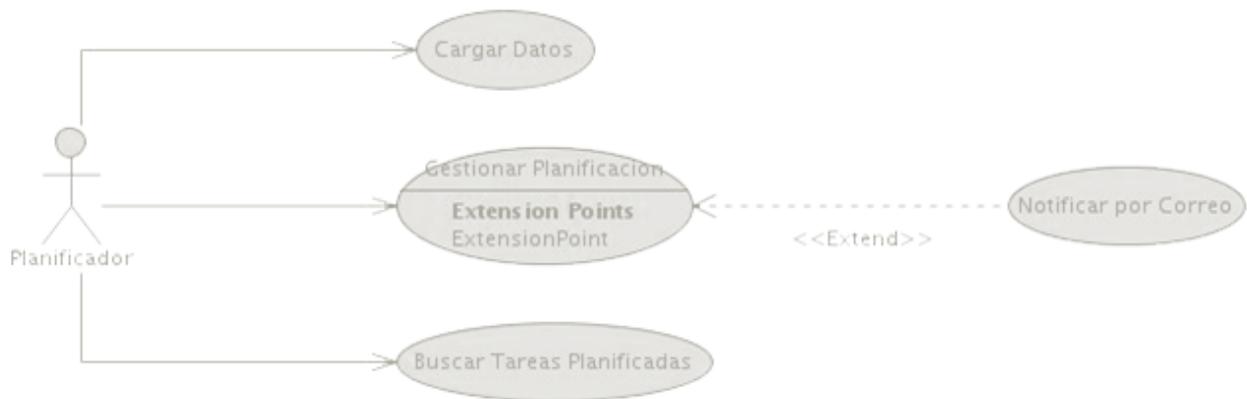


Figura 2 Diagrama de Casos de Uso del Sistema

3.6.3 Descripciones textuales de los Casos de Uso del Sistema

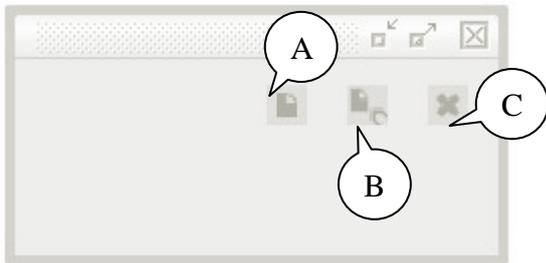
Caso de Uso	Cargar Datos.
Actores	Planificador
Resumen	El caso de uso comienza cuando el usuario configura el XML para cargar los recursos de las Bases de Datos del cliente para planificar las tareas y termina cuando se muestran los recursos en la aplicación de planificación.
Precondiciones	Debe existir conexión con la Base de Datos.
Referencias	RF1
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

	<ol style="list-style-type: none"> 1. El sistema lee el XML de configuración, verifica que su estructura sea correcta y que exista conexión a las diferentes Bases de Datos especificadas en el mismo.
	<ol style="list-style-type: none"> 2. El sistema muestra la interfaz de planificación con los recursos cargados de las Bases de Datos. <ul style="list-style-type: none"> • Se llena el árbol de recursos (A). • Se cargan los recursos en el campo seleccionable Recursos (B). • En caso de existir tareas planificadas, se cargan en el calendario de planificación.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. En caso de que el fichero de configuración XML 2. Si se configura de forma correcta el XML y no existe conexión con las Bases de Datos no se muestra ningún recurso en la aplicación de planificación y se muestra un mensaje de error que no existe conexión con la base de datos.
Poscondiciones	Se cargan los recursos existentes en las diferentes Bases de Datos del cliente especificadas en el XML para mostrarlos en la aplicación y realizar la planificación de tareas.

Tabla 2 Descripción textual del caso de Uso "Cargar Datos"

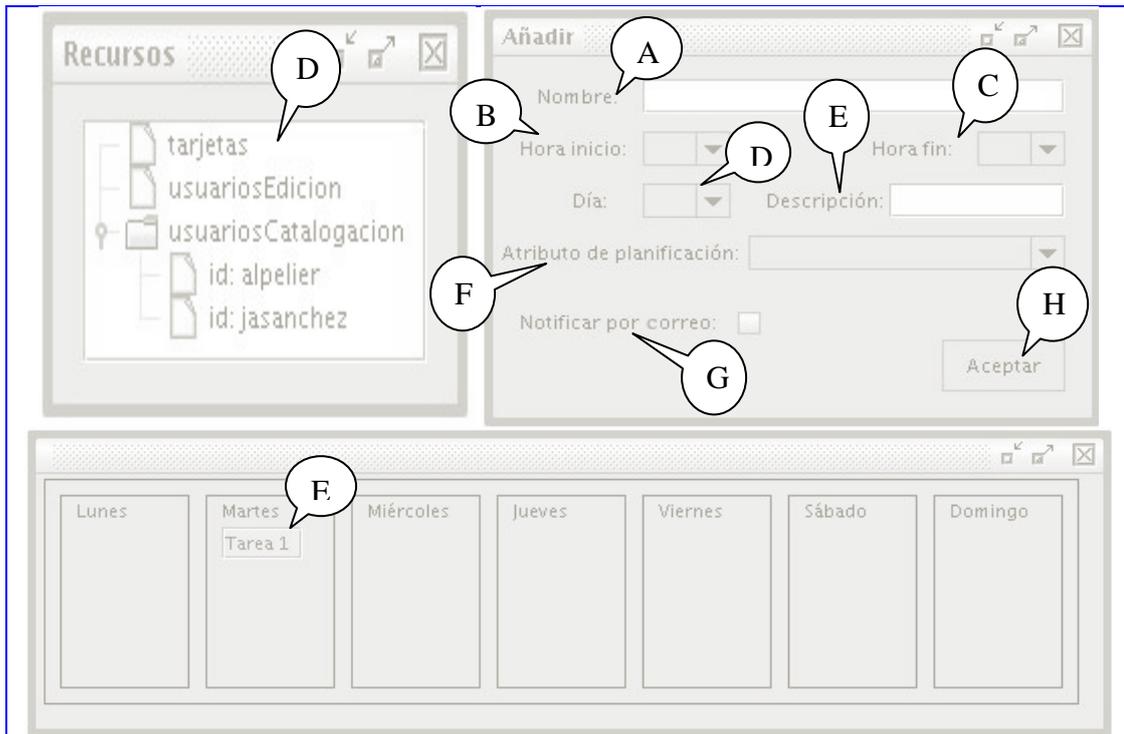
CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

Caso de Uso	Gestionar Planificación	
Actores	Planificador	
Resumen	El caso de uso comienza cuando el actor necesita gestionar la planificación para crear, modificar o eliminar una tarea y termina cuando se ha realizado una de estas acciones.	
Precondiciones	Debe estar conectado a la base de datos.	
Referencias	RF2, RF3, RF4	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
<p>1. El caso de uso comienza cuando el usuario selecciona una de las siguientes opciones:</p> <ul style="list-style-type: none"> • Crear Planificación (A): ver sección “Crear Planificación”. • Modificar Planificación (B): ver sección “Modificar Planificación”. • Eliminar Planificación (C): ver sección “Eliminar Planificación”. 		
Prototipo de Interfaz		
		
Sección “Crear Planificación”		
Acción del Actor	Respuesta del Sistema	

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

	2. El sistema muestra la lista de recursos disponibles para realizar las planificaciones de tareas. (D)
3. El usuario selecciona un recurso de la lista.	4. El sistema habilita el botón Insertar (A).
5. El usuario oprime el botón Insertar.	6. El sistema muestra la interfaz que posee los datos necesarios para crear una tarea.
7. El usuario llena los campos requeridos para crear la tarea: <ul style="list-style-type: none">• Nombre de la Planificación (A) (Obligatorio).• Hora Inicio (B) (Obligatorio).• Hora Fin (C) (Obligatorio).• Día (D) (Obligatorio).• Descripción (E) (Opcional).• Atributo de Planificación, específicos para el recurso seleccionado (F) (Obligatorio).• Notificar por correo (G) (Opcional). 8. El usuario oprime el botón Aceptar (H).	9. El sistema verifica que los datos estén correctos y muestra la tarea de planificación creada en el calendario de planificación (E). Culmina el Caso de Uso.
Prototipo de interfaz	

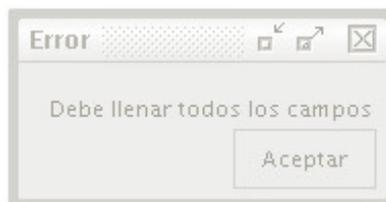
CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA



Flujos Alternos

Acción del Actor	Respuesta del Sistema
	7. Si se selecciona la opción Notificar por Correo ver Caso de Uso “Notificar por Correo”. Culmina el Caso de Uso.
	8. Si no introduce los datos obligatorios para la creación de una tarea, se muestra un mensaje de error: “Debe llenar todos los datos obligatorios”. Culmina el caso de Uso.

Prototipo de interfaz

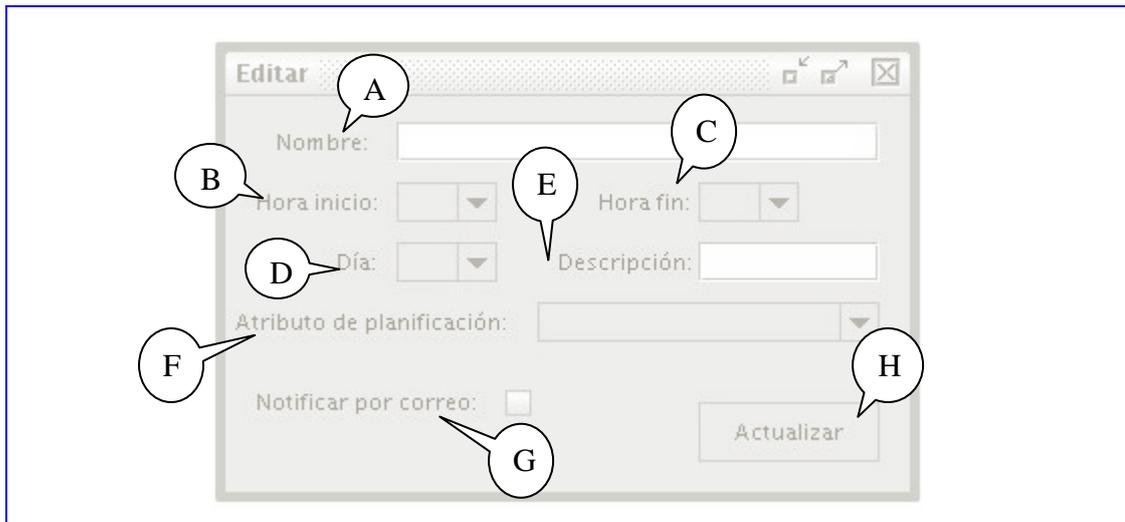


Sección “Modificar Planificación”

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

Acción del Actor	Respuesta del Sistema
<p>2. El usuario selecciona la tarea que desea modificar.</p>	<p>3. El sistema habilita el botón Modificar y muestra la interfaz para modificar los datos de la tarea seleccionada.</p>
<p>4. El usuario modifica los datos que desee, pudiendo ser:</p> <ul style="list-style-type: none"> • Nombre de la Planificación (A) (Obligatorio). • Hora Inicio (B) (Obligatorio). • Hora Fin (C) (Obligatorio). • Día (D) (Obligatorio). • Descripción (E) (Opcional). • Atributo de Planificación, específicos para el recurso seleccionado (F) (Obligatorio). • Notificar por correo (G) (Opcional). <p>5. El usuario oprime el botón Actualizar (H).</p>	<p>6. El sistema verifica que los datos estén correctos y actualiza los datos de la planificación modificada. Culmina el Caso de Uso.</p>
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>1. Si no está completo alguno de los datos obligatorios, se muestra un mensaje de error: “Debe llenar todos los datos”. Si se selecciona la opción Notificar por Correo ver Caso de Uso “Notificar por Correo”. Culmina el Caso de Uso.</p>
Prototipo de Interfaz	

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA



Sección “Eliminar Planificación”

Acción del Actor	Respuesta del Sistema
1. El usuario selecciona la tarea que desee eliminar.	2. El sistema habilita el botón Eliminar.
3. El usuario oprime el botón Eliminar o clic derecho/Eliminar.	4. El sistema muestra un mensaje de confirmación para eliminar la tarea.
5. El usuario confirma la acción.	6. El sistema elimina la tarea. Culmina el Caso de Uso.
Flujos Alternos	
5. El usuario cancela la eliminación de la tarea.	6. El sistema no elimina la tarea seleccionada y culmina el Caso de Uso.
Poscondiciones	Se gestiona la Planificación.

Tabla 3 Descripción textual del Caso de Uso "Gestionar Planificación"

Caso de Uso	Buscar Tareas Planificadas
Actores	Planificador
Resumen	El caso de uso comienza cuando el actor selecciona uno o los dos filtros de búsqueda de tareas planificadas, culmina cuando se muestran las tareas existentes en el sistema de acuerdo a los filtros introducidos.
Precondiciones	Debe existir conexión con la Base de Datos.

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

Referencias	RF5, RF6
Prioridad	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El Caso de Uso se inicia cuando el actor selecciona uno de los valores presentes en los campos de selección:</p> <ul style="list-style-type: none"> • Recursos (A): <ul style="list-style-type: none"> ○ Todos (aparece por defecto). ○ Tarjetas (opcional). ○ Usuarios de Edición (opcional). y/o • Fecha (B): <ul style="list-style-type: none"> ○ Semana (aparece por defecto). ○ Día (opcional). ○ Mes (opcional). 	<p>1. El sistema actualiza el calendario de tareas planificadas (C) dejando solo visibles las que cumplen con los parámetros seleccionados en los filtros de Recursos y/o Fecha y culmina el caso de uso.</p>
Prototipo de Interfaz	
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>1. Si no existe conexión con la Base</p>

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

	de Datos se muestra un mensaje de error informándolo. Culmina el Caso de Uso.
Poscondiciones	Se actualiza la vista de calendario de tareas planificadas de acuerdo a los filtros introducidos por el usuario.

Tabla 4 Descripción textual del Caso e Uso "Buscar tareas Planificadas"

Caso de Uso:	Notificar por Correo	
Actores:	Planificador	
Resumen:	El caso de uso comienza cuando se gestiona la planificación.	
Precondiciones:	Se debe realizar el caso de uso Gestionar Planificación.	
Referencias	RF 7	
Prioridad	Opcional	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Caso de Uso se inicia cuando se configura en el XML si se va a realizar o no la notificación.	2. El sistema muestra un checkbox para realizar la notificación.	
3. El usuario marca la opción Notificar por Correo.	4. El sistema envía una notificación al correo del usuario al cual le fue asignada la nueva tarea creada y/o modificada con todos los datos de la misma.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	1. Si en el XML se especifica que no desea notificar por correo no se muestra el checkbox.	
	3. Si no se marca el checkbox no se envía notificación al correo.	

CAPÍTULO 3: PRESENTACIÓN DE LA SOLUCIÓN PROPUESTA

Poscondiciones	Si se seleccionó el checkbox se envía una notificación al correo del usuario después de gestionada la tarea de planificación.
-----------------------	---

Tabla 5 Descripción textual del Caso de Uso "Notificar por Correo"

3.7 Conclusiones del capítulo

En este capítulo se describieron los casos de uso del sistema para así lograr una mejor comprensión de lo que el mismo debe realizar, para de esta forma lograr resultados satisfactorios. Además se tiene la descripción completa del sistema, obteniéndose aspectos como los requisitos funcionales y no funcionales del mismo. También se presenta un prototipo funcional de la aplicación de planificación.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.1 Introducción

En el presente capítulo se realiza la representación del diagrama de clases del diseño, dejando claridad del estilo arquitectónico a utilizar, se muestra además el diagrama de componentes y el modelo de despliegue. También se dejan expuestos los resultados obtenidos en las pruebas al módulo desarrollado.

4.2 Patrón arquitectónico

La elaboración de aplicaciones de mediano o gran tamaño requiere en gran medida de un correcto trabajo de la arquitectura de software, ya que permite descomponer el sistema en piezas que agrupan aspectos específicos del mismo y que al organizarse de cierta manera constituyen la base de la solución de un problema en particular. La arquitectura define la estructura y composición del sistema, cómo interactúan las partes que lo componen. De esta manera, la arquitectura de software puede ser vista como la estructura del sistema en función de la definición de los componentes y sus interacciones. (Bass, y otros, 2000)

El patrón de arquitectura establecido para el desarrollo en el proyecto es **Modelo Vista Controlador**. Es el más utilizado en aplicaciones web, ya que facilita la funcionalidad, mantenimiento y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite “no mezclar lenguajes de programación en el mismo código”. Este patrón independiza los datos de la aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos, lo que permite desarrollar o modificar por separado el modelo, la vista y la controladora sin afectar unas a otras. Básicamente la responsabilidad de cada parte es la siguiente:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista (Lescay, 2011).

En el módulo de planificación, la vista es una página HTML con contenido dinámico sobre la cual el usuario puede realizar acciones que son recogidas por el controlador, donde son transformadas en peticiones al modelo de acuerdo a la acción del usuario. Es usado este patrón para la arquitectura del módulo ya que reduce el esfuerzo en el momento del desarrollo del mismo, porque permite separar los componentes y así poder implementarlos por separado. Se propone el mismo MVC que presenta

symfony, ya que se realizó un análisis y no se requiere actualizar las vistas desde el modelo, además de tener en cuenta las características mencionadas anteriormente.

4.3 Patrones de diseño

La calidad de un sistema orientado a objetos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles. Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes, que resuelve un problema general de diseño en un contexto particular y tienden a ser independientes de los lenguajes y paradigmas de programación (Buschmann, y otros, 1996).

El patrón **Decorador** es utilizado porque adhiere dinámicamente funcionalidades adicionales a un objeto, proveyendo así una alternativa flexible a las subclasses para extender la funcionalidad, por ello cuando se desee cambiar de vistas en el calendario se muestran las tareas planificadas de acuerdo a la vista seleccionada, un ejemplo claro de esto es el layout, que es común para toda la aplicación.

También se ha tenido en cuenta el patrón **Controlador**, que propone que se debe asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente el sistema global. En la implementación del módulo se puede ver que la clase controladora PlanificacionActions se encarga de atender las solicitudes realizadas en el sistema.

4.4 Diagramas de Clases del Diseño

Los diagramas de clases del diseño constituyen un elemento esencial en la concepción de la aplicación que se propone, ya que servirán de guía a los desarrolladores para la construcción de una aproximación del sistema que se desea implementar, contribuyendo así a una mayor calidad del producto final (Lescay, 2011). El diagrama de clases del diseño muestra el sistema en términos de clases y métodos. Refleja el funcionamiento de la aplicación en términos lógicos (Hernández Moya, 2011). A continuación se muestran los diagramas de clases del diseño correspondientes para cada caso de uso del sistema.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

Notificar por Correo es el encargado de notificar al planificador o al usuario según el recurso seleccionado. Cuando se va a insertar o actualizar una tarea se muestra el formulario **EnviarCorreoForm**, este manda una petición por Ajax a la clase **PlanificacionActions**, donde se ejecuta el método **executeMandarCorreo()**, lo cual mediante el paquete Zend y las clases **Mail.php** y **Smtip.php** se envía el mismo.

4.4.3 Diagrama de clases del diseño Caso de Uso “Mostrar Planificación por Fecha”.

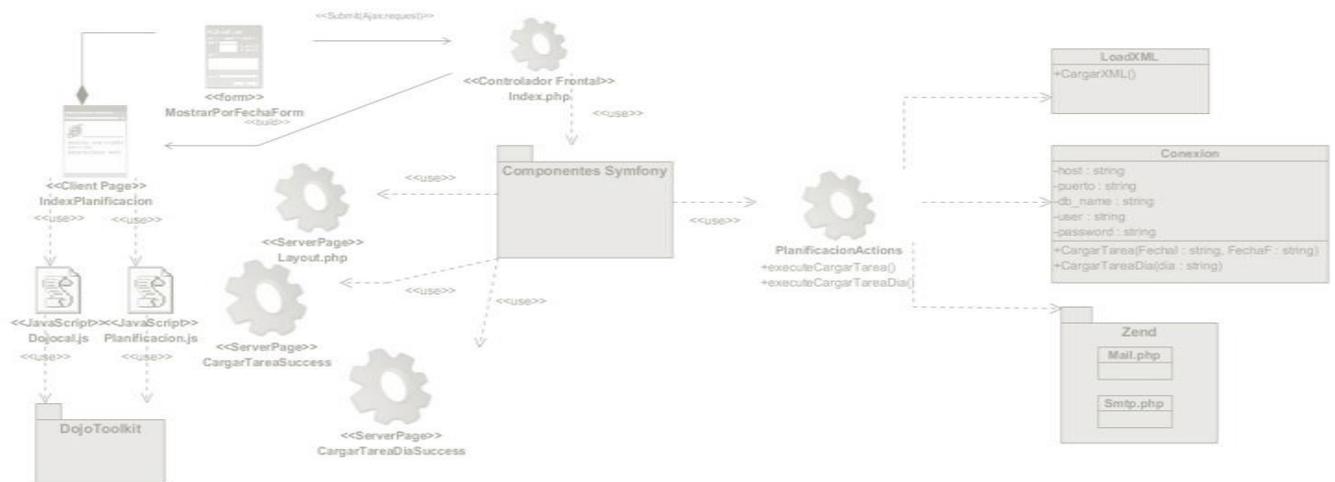


Figura 5 Diagrama de Clases del Diseño Caso de Uso "Mostrar Planificación por Fecha"

El caso de uso **Mostrar Planificación por Fecha** describe el proceso del sistema ante la opción seleccionada por el planificador. Se escoge la opción en el formulario **MotrarPorFechaForm**, el cual manda una petición por Ajax a la clase **PlanificacionActions** donde se ejecuta la funcionalidad **executeCargarTarea()** o **executeCargarTareaDia()**, según la opción seleccionada, en el cual se accede a la base de datos mediante la clase conexión y se buscan las tareas según los parámetros.

4.4.4 Diagrama de clases del diseño Caso de Uso “Filtrar por Recurso”

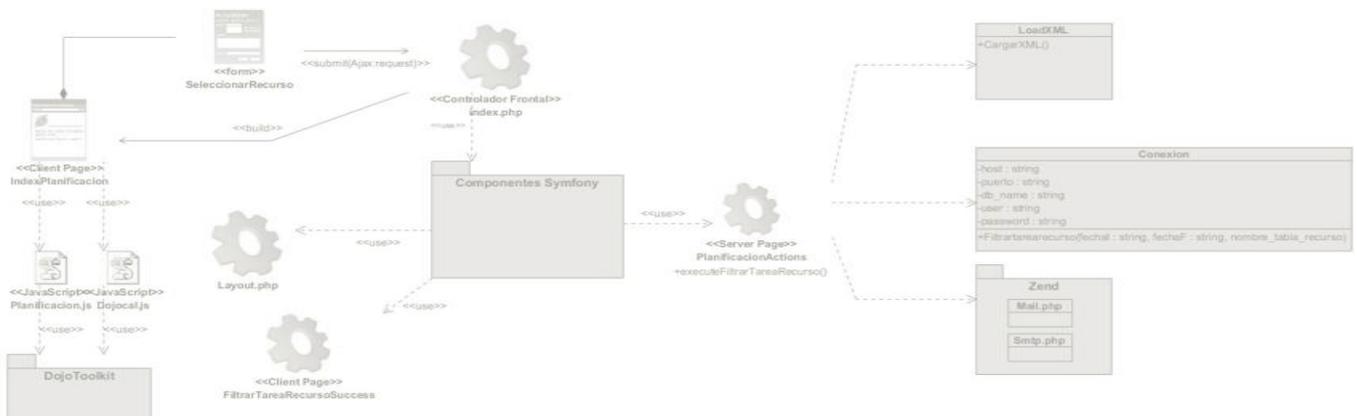


Figura 6 Diagrama de Clases del Diseño Caso de Uso "Filtrar por Recurso"

El caso de uso Filtrar por Recurso describe el proceso del sistema ante la opción seleccionada por el planificador de filtrar las tareas según el recurso. El mayor peso recae sobre planificacion.js, la cual es la encargada de tomar el valor seleccionado en el formulario **SeleccionarRecurso** y mandar la petición por Ajax a la clase **PlanificacionActions**, donde se ejecuta el método **executeFiltrarTareaRecurso()**, donde este se encarga de mandar a hacer una búsqueda a la base de datos mediante la clase conexión.

4.4.5 Diagrama de clases del diseño Caso de Uso "Cargar Datos"

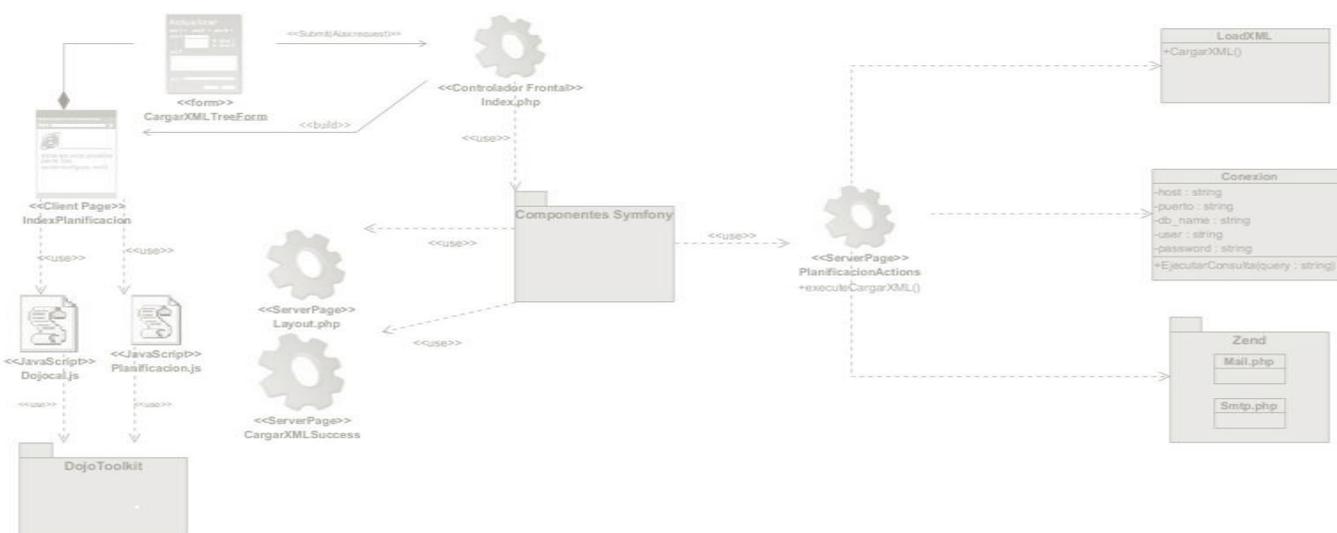


Figura 7 Diagrama de Clases del Diseño Caso de Uso "Cargar Datos"

Cargar Datos es el encargado de leer el XML y según este, mostrar en la vista el tree de recursos. Cuando se ejecute el método **add_onload()** de planificacion.js, este manda una petición por Ajax a la clase **PlanificacionActions**, donde se ejecuta el método **executeCargarXML()**, lo cual mediante la clase del modelo **LoadXML**, lee el archivo XML.

4.5 Estándar de codificación

Para lograr un mejor entendimiento del código de cualquier programa es necesario realizar los diferentes diagramas expuestos anteriormente, para así asegurar que el código salga con la calidad requerida y sea fácil de entender y actualizar, ya que es construido por un grupo de programadores. En el desarrollo del módulo de planificación se estableció el siguiente estándar.

General: Para archivos que contengan únicamente código PHP, la etiqueta de cierre (">") no está permitida. No es requerida por PHP, y omitirla evita la inyección de espacios en blanco en la respuesta.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

Clases: Las clases solo van a contener nombres alfanuméricos, aunque no es aconsejable el uso de los números. Los guiones bajos son usados como separador de ruta. Si las clases están formadas por más de una palabra, la primera letra de cada palabra debe aparecer en mayúsculas.

Funciones y Métodos: Los nombres de las funciones deben empezar siempre con letra minúscula. Los guiones bajos (_) solo están permitidos para el caso en que el nombre de la función tenga más de una palabras. Después del guion bajo (_) debe comenzar con mayúscula solo la primera letra. Los números están permitidos en los nombres de función pero no se aconseja en la mayoría de los casos. Por norma general, se recomienda la elocuencia. Los nombres de función deben ser lo suficientemente elocuentes como para describir su propósito y comportamiento.

Variables: Los nombres de las variables deben empezar siempre con una letra en minúscula. Los guiones bajos (_) solo están permitidos para el caso en que el nombre de la variable tenga más de una palabra y después del guion se mantiene en minúscula. Por norma general, se recomienda la elocuencia. Las variables deberán ser siempre tan elocuentes como prácticas para describir los datos que el desarrollador pretende almacenar en ellas. Variables escritas como " \$i " y " \$n " no son aconsejables, salvo para el contexto de los bucles más pequeños. Si un bucle contiene más de 20 líneas de código, las variables de índice deberían tener nombres más descriptivos.

4.6 Modelo de implementación

El modelo de implementación está compuesto por un conjunto de subsistemas y componentes que establecen la composición física de la implementación del sistema, tiene como objetivos implementar las clases de diseño como componentes, asignar los componentes a los nodos³, probar los componentes individualmente, integrar los componentes en un sistema ejecutable. Utiliza los Diagramas de Componentes y Diagrama de Despliegue para comprender cómo se organizan y dependen unos de otros (Jacobson, et al., 2000).

4.7 Diagrama de Despliegue

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación que muestra las relaciones físicas entre los componentes hardware y software en el sistema propuesto (Fernández Vilas, 2010). A continuación se muestra el diagrama de despliegue que corresponde al módulo de planificación.

³ **Nodos:** Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

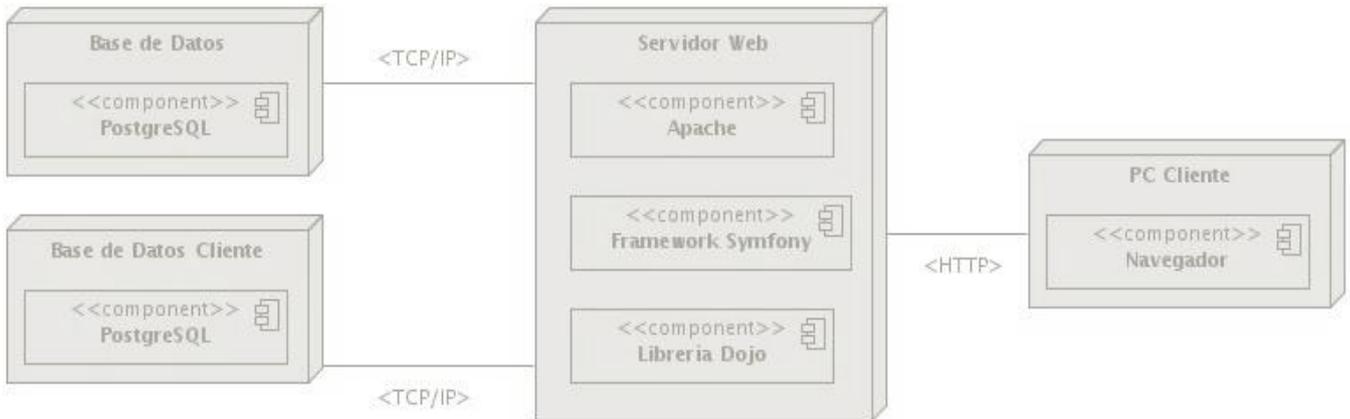


Figura 8 Diagrama de Despliegue

4.8 Diagrama de componentes

Los diagramas de componentes son utilizados para lograr una estructura en el modelo de implementación y mostrar las relaciones entre sus elementos. Además, los mismos muestran los componentes de software que constituyen una parte reusable, sus interfaces y sus interrelaciones. A continuación se muestra el diagrama que corresponde al módulo desarrollado.

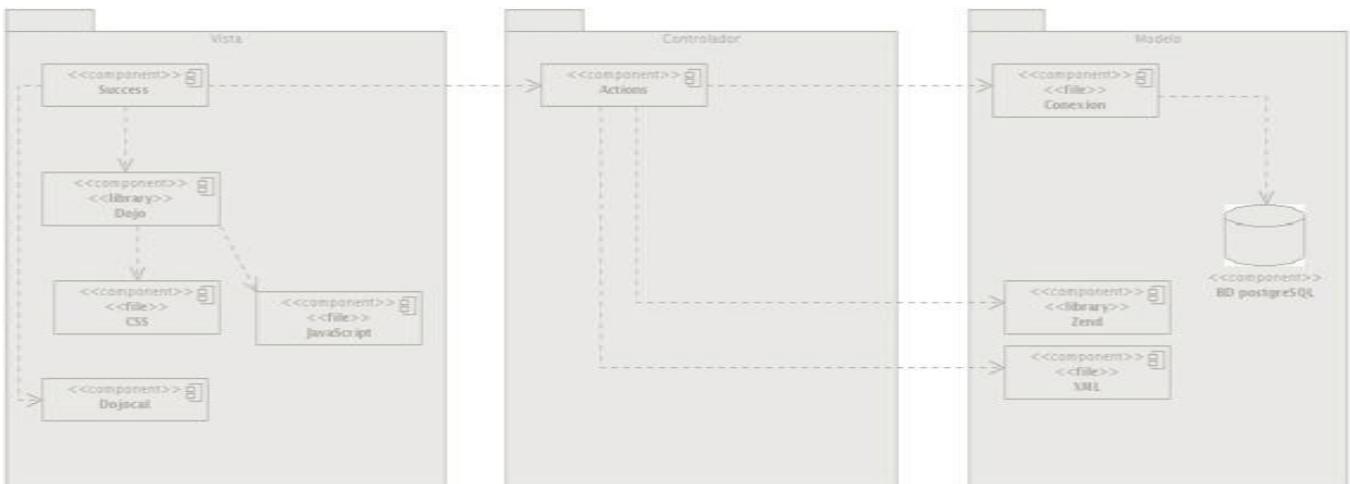


Figura 9 Diagrama de Componentes

4.9 Pruebas realizadas al módulo

4.9.1 Pruebas de Caja Negra

Se decide realizar este método de prueba porque el mismo se refiere a las pruebas aplicadas sobre la interfaz de la aplicación, las cuales se centran en los requisitos funcionales del software y dicha prueba pretende demostrar que las funciones se cumplen y se espera un resultado correcto.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.9.2 Casos de Prueba

Un caso de prueba permite detallar la ejecución de las pruebas, incluyendo los datos de entrada, las condiciones de ejecución y los resultados obtenidos.

Se debe verificar que:

- ✓ El módulo cumpla con los requerimientos descritos en la especificación de requisitos.
- ✓ Si los resultados son los esperados, y la aplicación se comporta tal y como se describe en la especificación de los casos de uso.

A continuación se presenta un resumen de las pruebas realizadas a la aplicación:

4.9.2.1 Caso de Prueba para el Caso de Uso “Cargar Datos”

Sección	Escenario	Entrada	Resultados Esperados	Respuesta del Sistema	Respuesta de la prueba
SC 1 Cargar datos.	EC 1.1 Cargar Datos exitosamente.	El usuario desea ver la aplicación de planificación con los recursos cargados de las Bases de Datos del Cliente y para ello configura el XML.	El sistema muestra la aplicación con los recursos cargados de las Bases de Datos.	Se muestra la aplicación con los recursos cargados previamente.	Se escribe el resultado que se obtiene al realizar la prueba. Ej.: Satisfactorio.
	EC 1.2 Cargar Datos falló porque no se conectó con la Base de Datos.	El usuario desea ver la aplicación de planificación con los recursos cargados de la Base de	El sistema muestra un mensaje de error.	Se muestra el mensaje.	Satisfactorio.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

		Datos del Cliente y para ello configura el XML.			
--	--	---	--	--	--

Tabla 6 Caso de Prueba para el Caso de Uso "Cargar Datos"

4.9.2.2 Caso de Prueba para el Caso de Uso "Gestionar Planificación"

Sección	Escenario	Entrada	Resultados Esperados	Respuesta del Sistema	Respuesta de la prueba
SC 1 Insertar tarea.	EC 1.1 Insertar tarea exitosamente.	El usuario desea insertar una tarea, para ello llena los datos en la interfaz, pudiendo ser Nombre, Hora Inicio, Hora Fin, Descripción, Día y si desea Notificar por Correo y oprime el botón Insertar.	El sistema debe mostrar una ventana con los datos necesarios para insertar una tarea de planificación.	Se muestra la ventana con los datos para la inserción de la tarea y se inserta.	Se escribe el resultado que se obtiene al realizar la prueba. Ej.: Satisfactorio.
	EC 1.2 Insertar tarea de forma incorrecta.	El usuario desea insertar una tarea, para ello llena los datos en la interfaz, pudiendo ser Nombre, Hora Inicio, Hora Fin, Descripción, Día	El sistema debe mostrar un mensaje indicando que deben llenarse los campos requeridos.	Se muestra un mensaje de error	En una primera iteración de las pruebas fue No Satisfactorio, pero luego se corrigieron los errores.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

		y si desea Notificar por Correo y oprime el botón y oprime el botón Insertar.			
SC 2 Modificar tarea.	EC 2.1 Modificar tarea exitosamente.	El usuario desea modificar una tarea, para ello llena los datos en la interfaz, pudiendo ser Nombre, Hora Inicio, Hora Fin, Descripción, Día y si desea Notificar por Correo y oprime el botón y oprime el botón Editar o da clic derecho sobre la misma.	El sistema debe mostrar una ventana con los datos necesarios para modificar una tarea de planificación.	Se muestra la ventana con los datos para la modificación de la tarea y se modifica.	Satisfactorio.
	EC 2.2 Modificar tarea de forma incorrecta.	El usuario desea modificar una tarea, para ello llena los datos en la interfaz, pudiendo ser Nombre, Hora Inicio, Hora Fin, Descripción, Día y si desea Notificar por Correo y oprime	El sistema debe mostrar un mensaje indicando que deben llenarse los campos requeridos.	Se muestra un mensaje de error.	Satisfactorio.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

		el botón y oprime el botón Editar o da clic derecho sobre la misma.			
	EC 2.3 Modificar tarea falló porque no se seleccionó la tarea.	El usuario desea modificar una tarea, para ello llena los datos en la interfaz, pudiendo ser Nombre, Hora Inicio, Hora Fin, Descripción, Día y si desea Notificar por Correo y oprime el botón y oprime el botón Editar o da clic derecho sobre la misma.	El sistema envía un mensaje indicando que debe seleccionarse una tarea.	Se muestra un mensaje de error.	Satisfactorio.
SC 3 Eliminar tarea.	EC 3.1 Eliminar tarea exitosamente.	El usuario desea eliminar una tarea, para ello escoge la tarea que desee y oprime el botón Eliminar o da clic derecho Eliminar.	El sistema muestra la aplicación con los datos actualizados, o sea, con la tarea eliminada.	Se elimina la tarea seleccionada	Satisfactorio.
	EC 3.2 Eliminar tarea falló porque no se seleccionó la tarea.	El usuario desea eliminar una tarea, para ello oprime el botón Eliminar.	El sistema envía un mensaje indicando que debe	Se muestra un mensaje de error.	Satisfactorio.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

			seleccionar una tarea.		
--	--	--	------------------------	--	--

Tabla 7 Caso de Prueba para el Caso de Uso "Gestionar Planificación"

4.9.2.3 Caso de Prueba para el Caso de Uso "Buscar Tareas Planificadas"

Sección	Escenario	Entrada	Resultados Esperados	Respuesta del Sistema	Respuesta de la prueba
SC 1 Mostrar Planificación por Fecha.	EC 1.1 Mostrar Planificación dada una fecha seleccionada.	El usuario desea ver las planificaciones realizadas por día, semana o mes.	El sistema muestra la aplicación con las tareas planificadas de acuerdo a la fecha escogida.	Se muestran las tareas correspondientes a la fecha seleccionada	Se escribe el resultado que se obtiene al realizar la prueba. Ej.: Satisfactorio.
	EC 1.2 Mostrar Planificación dado un recurso seleccionado.	El usuario desea filtrar por las tareas que corresponden a un recurso dado.	El sistema muestra la aplicación con las tareas que corresponden al recurso seleccionado en el filtro.	Se muestran las tareas correspondientes a dicho recurso.	Satisfactorio.
	EC 1.3 Mostrar Planificación dado un recurso y una fecha dada.	El usuario desea ver las planificaciones correspondientes a un recurso en una fecha determinada.	El sistema muestra la aplicación con las tareas planificadas para el recurso seleccionado de acuerdo a la fecha escogida.	Se muestran las tareas correspondientes al recurso en una fecha determinada, aunque por defecto aparece por semana.	Satisfactorio.

Tabla 8 Caso de Prueba para el Caso de Uso "Buscar Tareas Planificadas"

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

4.9.2.7 Caso de Prueba para el Caso de Uso “Notificar por Correo”

Sección	Escenario	Entrada	Resultados Esperados	Respuesta del Sistema	Respuesta de la prueba
SC Notificar por correo	EC 1 Notificar por correo exitosamente	El usuario selecciona la opción Notificar por Correo.	El sistema envía una notificación al correo del usuario, dependiendo el recurso seleccionado.	Se envía la notificación al correo del usuario, dependiendo el recurso seleccionado.	Se escribe el resultado que se obtiene al realizar la prueba. Ej.: Satisfactorio.
	EC 1.2 Notificar por Correo falló porque no existe conexión con la Base de Datos.	El usuario selecciona la opción Notificar por Correo.	El sistema no envía la notificación al correo.	No se envía ninguna notificación.	Satisfactorio.

Tabla 9 Caso de Prueba para el Caso de Uso "Notificar por Correo"

4.10 Resultados de las pruebas

Las pruebas de caja negra se realizaron estrictamente de acuerdo a los casos de prueba elaborados. Al finalizar las mismas se detectó que, en una primera iteración, existió una funcionalidad que no realizaba la acción esperada. La no conformidad referente a Insertar Tarea, fue corregida y actualmente devuelve los resultados esperados.

4.11 Conclusiones del capítulo

En este capítulo se presentaron los artefactos a tener en cuenta para el desarrollo de la aplicación web de planificación. Se exponen además los diagramas de clases del diseño, diagrama de componentes, diagrama de despliegue y el modelo de implementación. Para ello se explican las razones del uso del patrón de arquitectura Modelo-Vista-Controlador y los patrones de diseño. También se planteó el estándar de codificación, el cual ayuda a tener un mejor entendimiento del código del módulo desarrollado. Con el resultado de las pruebas realizadas se garantiza que la aplicación cumple con las funcionalidades especificadas.

CONCLUSIONES GENERALES

Concluida la realización de este trabajo de diploma se comprobó que la planificación de tareas es necesaria para una mayor calidad con el trabajo con materiales audiovisuales. Con el desarrollo de la aplicación se establecieron los siguientes resultados que avalan el cumplimiento del objetivo trazado con anterioridad:

- ✓ El estudio de otras soluciones a nivel mundial no demostró que todos los sistemas de gestión de medias integran las funcionalidades necesarias para la planificación de tareas con materiales audiovisuales, porque fue imposible obtener la información teniendo en cuenta de que son propietarios.
- ✓ El proceso de desarrollo del software realizado permitió generar todos los artefactos y documentación correspondientes al mismo a partir de la metodología identificada, lo que sienta las bases para el desarrollo del producto final que desea desplegar en el proyecto SCCM.
- ✓ El uso de tecnologías y herramientas libres para el desarrollo del módulo permitió cumplir con la política de soberanía tecnológica por la que aboga el país.
- ✓ A partir de las pruebas realizadas se garantizó que el sistema obtenido cumple satisfactoriamente con las funcionalidades especificadas.

Por todo lo descrito se demostró el cumplimiento de los objetivos de la investigación, a través de los requisitos funcionales y diseño arquitectónico, lo cual favoreció a la construcción de la aplicación de forma robusta y eficiente.

RECOMENDACIONES

Para darle continuidad al presente trabajo e incrementar las prestaciones del mismo se recomienda:

- ✓ Investigar las formas de comunicación con otros sistemas para garantizar la integración con los productos del departamento.
- ✓ Investigar y añadir funcionalidades al sistema en función de los nuevos requisitos que puedan ser identificados.

REFERENCIA BIBLIOGRÁFICA

- López, Gustavo Romero. 2008.** [Online] 2008. [Cited: 6 6, 2012.] <http://geneura.ugr.es/~gustavo/xampp/php.shtml>.
- EIPais. 2011.** EIPais. *EIPais*. [Online] 2011. <http://servicios.elpais.com/diccionarios/castellano/editar>.
- WordPress. 2010.** Definicion.de. *Definicion.de*. [Online] 2010. [Cited: 02 17, 2012.] <http://definicion.de/html/>
- Carrión, Daniela. 2008.** *Manual de JAVASCRIPT*. s.l. : Universidad del Azuay, 2008.
- SANTIAGO, CARLOS ALBERTO COVARRUBIAS. 2012.** [Online] 2012. [Cited: 6 6, 2012.] <http://dwlcj.bligoo.com.mx/tarea-2>.
- Pérez, Javier Eguíluz. 2008.** libros Web. *Introducción a AJAX*. [Online] 2008. [Cited: 2 17, 2012.] www.librosweb.es.
- Pérez, Jeam Michael Suárez. 2009.** *Sistema de captura e indexación de video*. La habana : Tesis, 2009.
- Fabien Potencier, François Zaninotto. 2009.** *Symfony la guía definitiva Doctrine*. s.l. : Apress, 2009. 978-1590597866.
- Eguiluz, Javier. 2012.** *Symfony.es*. *Symfony.es*. [Online] 2012. <http://www.symfony.es/que-es-symfony/>.
- Russell, Matthew A. 2008.** *Dojo The Definitive Guide*. s.l. : Orreilly, 2008. 9780596516482.
- CAJAS, MARIA DEL CARMEN ERAZO. 2007.** *DISEÑO E IMPLEMENTACIÓN DE SEGURIDADES LÓGICAS EN REDES INALÁMBRICAS DE DATOS FIJAS EN LOS LABORATORIOS DE LA FACULTAD DE SISTEMAS E INFORMÁTICA DE LA ESPE-L*. Ecuador : ESCUELA POLITÉCNICA DEL EJÉRCITO SEDE LATACUNGA, 2007.
- González, Carlos D. 2012.** *Curso PostgreSQL, SQL avanzado y PHP*. *Curso PostgreSQL, SQL avanzado y PHP*. [Online] 2012. [Cited: 2 16, 2012.] <http://www.usabilidadweb.com.ar/postgre.php>.
- Martinez, Rafael. 2010.** PostgreSQL-es. *PostgresSQL-es*. [Online] 2010. [Cited: 2 16, 2012.] http://www.postgresql.org.es/sobre_postgresql.
- Teleformacion. 2012.** Teleformacion. *Teleformacion*. [Online] 4 1, 2012. <http://eva.uci.cu>.
- Telefomación. 2008.** Telefomación. *Telefomación*. [Online] 2008. [Eva.uci.cu](http://eva.uci.cu). 2009Teleformaciónhttp://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Complementarios/Materiales_Complementarios_Conf_1/UML.pdf
- Larman, Craig. 2008.** *Introducción al Análisis y Diseño Orientado a Objetos*. Uruguay : s.n., 2008.
- Figueroa, P. 2007.** Conceptos básicos en un Diagrama de Colaboración, 2005. *Conceptos básicos en un Diagrama de Colaboración, 2005*. [Online] 2007. <http://www.cs.ualberta.ca/~pfiguero/soo/uml/colaboracion01.html>.
- Oriente, Universidad de. 2008.** Ingeniería de software. *tema: Capturar los requerimientos de comportamiento del sistema*. [Online] 2008. [Cited: 6 6, 2012.] http://www.moodle.uho.edu.cu/.../Ingenieria_de_software_2da_conferencia_de_....
- Fernández Vilas, Ana. 2010.** Laboratorio de Televisión Digital Interactiva. *Laboratorio de Televisión Digital Interactiva*. [Online] 2010. <http://tvdi.det.uvigo.es/~avilas/UML/node50.html>.
- Bass, L., Clements, P. and Kasman, R. 2000.** *Architecture in practice*. s.l. : Addison-Wesley, 2000. 0201199300.
- Lescay, Roberto Frómata. 2011.** *Subsistema de Control de Ingesta del Sistema*. La Habana : s.n., 2011.

BIBLIOGRAFÍA

- Alvarez, Rubén. 2001.** DesarrolloWeb.com. *DesarrolloWeb.com*. [Online] Septiembre 24, 2001. <http://www.desarrolloweb.com/articulos/534.php>.
- Antonio Goñi, Jose R. Zubizarreta. 2006.** Ingeniería del Software. *Ingeniería del Software*. [Online] 2006. [Cited: 05 13, 2011.] <http://es.scribd.com/doc/49666181/19/Prueba-de-integracion-ascendente-de-SiguientePrimo-y-EsPrimo>.
- Aranda, Alex. 2010.** aTunes. *aTunes*. [Online] 2010. [Cited: 05 12, 2011.] http://www.atunes.org/?page_id=7.
- Bass, L., Clements, P. and Kasman, R. 2000.** *Architecture in practice*. s.l. : Addison-Wesley, 2000. 0201199300.
- Buchmann, Frank. 1996.** *Pattern Oriented Software Architecture 1 edition* . 1996.
- Buschmann, F., et al. 1996.** *Pattern – Oriented Software Architecture. A System of Patterns*. 1996. 978-0471958697.
- CAJAS, MARIA DEL CARMEN ERAZO. 2007.** *DISEÑO E IMPLEMENTACIÓN DE SEGURIDADES LÓGICAS EN REDES INALÁMBRICAS DE DATOS FIJAS EN LOS LABORATORIOS DE LA FACULTAD DE SISTEMAS E INFORMÁTICA DE LA ESPE-L*. Ecuador : ESCUELA POLITÉCNICA DEL EJÉRCITO SEDE LATACUNGA, 2007.
- Carrión, Daniela. 2008.** *Manual de JAVASCRIPT*. s.l. : Universidad del Azuay, 2008.
- Eguiluz, Javier. 2012.** Symfony.es. *Symfony.es*. [Online] 2012. <http://www.symfony.es/que-es-symfony/>.
- EIPaís. 2011.** EIPais. *EIPais*. [Online] 2011. <http://servicios.elpais.com/diccionarios/castellano/editar>.
- Escovar, Marcos. 2003.** Desarrollo Web.com. *Desarrollo Web.com*. [Online] 2003. <http://www.desarrolloweb.com/articulos/1545.php>.
- Fabián, Orlando. 2005.** DesarrolloWeb.com. *DesarrolloWeb.com*. [Online] Marzo 3, 2005. <http://www.desarrolloweb.com/articulos/1853.php>.
- Fabien Potencier, François Zaninotto. 2009.** *Symfony la guía definitiva Doctrine*. s.l. : Apress, 2009. 978-1590597866.
- Fernández Vilas, Ana. 2010.** Laboratorio de Televisión Digital Interactiva. *Laboratorio de Televisión Digital Interactiva*. [Online] 2010. <http://tvdi.det.uvigo.es/~avilas/UML/node50.html>.
- Figueroa, P. 2007.** Conceptos básicos en un Diagrama de Colaboración, 2005. *Conceptos básicos en un Diagrama de Colaboración, 2005*. [Online] 2007. <http://www.cs.ualberta.ca/~pfiguero/soo/uml/colaboracion01.html>.
- Florencia. 2010.** definicionabc. [Online] 01 05, 2010. [Cited: 02 20, 2011.] <http://www.definicionabc.com/general/fotograma.php>.
- Fowler, Martin. 2004.** *UML distilled: " A brief Guide to the Standard Object Modeling Lenguaje "*. 2004. ISBN 020189151X.
- Full, Ingeniería. 2008.** Ingeniería Full. *Ingeniería Full*. [Online] 2008. <http://ingenieria.programasfull.com/descargar-tutorial-programacion-con-lenguaje-php.html>.
- Gamma. 1995.** *Design Patterns*. 1995.
- González, Carlos D. 2012.** Curso PostgreSQL, SQL avanzado y PHP. *Curso PostgreSQL, SQL avanzado y PHP*. [Online] 2012. [Cited: 2 16, 2012.] <http://www.usabilidadweb.com.ar/postgre.php>.
- Gracia, Joaquin. 2003.** Análisis y Diseño. [Online] 09 27, 2003. [Cited: 02 17, 2011.] <http://www.ingenierosoftware.com/analisisydiseño/casosdeuso.php>.
- Gutierrez, Jorge A Saavedra. 2007.** El mundo Informatico. *El mundo Informatico*. [Online] 05 05, 2007. [Cited: 04 13, 2011.] <http://jorgesaaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>.

BIBLIOGRAFÍA

- Hernández Moya, Iván. 2011.** *Migración del Módulo de Catalogación de materiales audiovisuales del producto Captura y catalogación de medias.* La Habana : Universidad de las Ciencias Informáticas, 2011.
- Informatica, HCA. 2011.** [Online] Agosto 2011.
http://hcainformatica.blogspot.com/2011_08_01_archive.html.
- International, Visual Paradigm. 1999.** *Visual Paradigm. Visual Paradigm.* [Online] 1999. [Cited: 1 28, 2012.] <http://www.visual-paradigm.com/product/vpuml/communityedition.jsp>.
- Ivar Jacobson, Grady Booch, James Rumbaugh.** *El proceso unificado de desarrollo de software.* s.l. : Addison Wesley.
- Jacobson, Ivar. 2000.** *El proceso unificado de desarrollo de software.* Madrid : Pearson Educación, 2000. 8478290362.
- James Rumbaugh, Ivar Jacobson, Grady Booch.** *El lenguaje de modelado unificado. Manual de referencia.* s.l. : Addison Wesley.
- Larman, Craig. 2008.** *Introducción al Análisis y Diseño Orientado a Objetos.* Uruguay : s.n., 2008.
- Lescay, Roberto Frómata. 2011.** *Subsistema de Control de Ingesta del Sistema.* La Habana : s.n., 2011.
- librosweb.** librosweb.es. *librosweb.es.* [Online] [Cited: 02 17, 2012.]
<http://www.librosweb.es/javascript/capitulo1.html>.
- López, Gustavo Romero. 2008.** [Online] 2008. [Cited: 6 6, 2012.]
<http://geneura.ugr.es/~gustavo/xampp/php.shtml>.
- Martínez Fajardo, Alexander. 2010.** *tallertematico2010. tallertematico2010.* [Online] 12 8,9,10, 2010.
<http://tallertematico2010.fordes.co.cu/public/site/18.pdf>.
- Martinez, Rafael. 2010.** *PostgreSQL-es. PostgreSQL-es.* [Online] 2010. [Cited: 2 16, 2012.]
http://www.postgresql.org.es/sobre_postgresql.
- Molpeceres, Alberto. 2002.** *Procesos de desarrollo: RUP,XP y FDD.* 2002.
- Oracle corporation. 2012.** *NetBeans. NetBeans.* [Online] 2012. [Cited: 2 9, 2012.]
http://netbeans.org/index_es.html.
- Oriente, Universidad de. 2008.** *Ingeniería de software. tema: Capturar los requerimientos de comportamiento del sistema.* [Online] 2008. [Cited: 6 6, 2012.]
http://www.moodle.uho.edu.cu/.../Ingenieria_de_software_2da_conferencia_de_....
- Peña, Ing. Yadira Machado. 2009.** *Estrategia de Pruebas de función .* 2009.
- Pérez, Javier Eguíluz. 2008.** *libros Web. Introducción a AJAX.* [Online] 2008. [Cited: 2 17, 2012.]
www.librosweb.es.
- Pérez, Jeam Michael Suárez. 2009.** *Sistema de captura e indexación de video.* La habana : Tesis, 2009.
- Pérez, Yaquelin de la Caridad Tejeda. 2010.** *Implementación del subsistema Gestión y Presentación de Contenido de la Plataforma VideoWeb.* Ciudad de la Habana : s.n., 2010.
- Piñero, Rafael. 2007.** *visualbeta. visualbeta.* [Online] 11 29, 2007. [Cited: 12 14, 2010.]
<http://www.visualbeta.es/3009/software-libre/vlc-el-mejor-reproductor-multimedia/>.
- Ramos, Maylen Ortiz. 2008.** *Análisis y Diseño de un sistema para la planificación automatizada del Horario Docente de la facultad 4.* La Habana : Tesis, 2008.
- Ray Edmondson. 1998.** *FILOSOFIA DE LOS ARCHIVOS AUDIOVISUALES.* [Online] 06 1998. [Cited: 04 12, 2011.] <http://filosofadelosarchivosaudiovisuales.blogspot.com/>.
- Real academia española.** *Real academia española. Real academia española.* [Online] Real academia española. [Cited: 11 7, 2011.] <http://buscon.rae.es/drael/>.
- RibosoMatic. 2010.** *RibosoMatic, desarrollo web y tecnología. RibosoMatic, desarrollo web y tecnología.* [Online] 2010. <http://www.ribosomatic.com/articulos/implementar-xml-rpc-en-php/>.
- Russell, Matthew A. 2008.** *Dojo The Definitive Guide.* s.l. : Orreilly, 2008. 9780596516482.
- Sande, Martin. 2004.** *Programación en C++ con Qt bajo Entorno GNU/Linux.* 2004.

BIBLIOGRAFÍA

- SANTIAGO, CARLOS ALBERTO COVARRUBIAS. 2012.** [Online] 2012. [Cited: 6 6, 2012.] <http://dwlcj.bligoo.com.mx/tarea-2>.
- SOMMERVILLE, IAN. 2005.** *Ingeniería de Software. Séptima edición.* España : s.n., 2005.
- Teleformación. 2008.** Teleformación. *Teleformación.* [Online] 2008. [Eva.uci.cu](http://eva.uci.cu).
- 2009**Teleformaciónhttp://teleformacion.uci.cu/file.php/102/Curso_2008-2009/Materiales_Complementarios/Materiales_Complementarios_Conf_1/UML.pdf
- 2009**Teleformación<http://teleformacion.uci.cu/mod/resource/view.php?id=11361>
- Teleformacion. 2012.** Teleformacion. *Teleformacion.* [Online] 4 1, 2012. <http://eva.uci.cu>.
- Together, Borland. 2010.** SlideShare Inc. *SlideShare Inc.* [Online] 2010. [Cited: 1 28, 2012.] <http://www.slideshare.net/unimauro/borland-together-administration-bases-de-datos>.
- Torres, Youssary Hernández. 2009.** *Módulos del Dotproject para la gestión de recursos y horarios.* La Habana : Tesis, 2009.
- UCI,Colectivo de autores.** Conferencia 7:Disciplina Modelamiento del negocio. Modelo de Negocio. Modelo de Dominio, Modelo conceptual. DCUN, D, Actividades y Doc. Visión. . *Fase de Inicio. Disciplina de Modelamiento del negocio.*
- UCI.Colectivo de Autores. 2007.** Conferencia 2: Ingeniería de Software. 2007.
- Uyaguay León, Ana Lucía Yunga Barros,Myriam Alexandra. 2010.** Repositorio Digital. *Repositorio Digital.* [Online] 10 8, 2010. [Cited: 12 14, 2010.] <http://dspace.ups.edu.ec/handle/123456789/316>.
- VIDEOMA. 2009.** ISID. *ISID.* [Online] 2009. [Cited: 2 15, 2012.] http://www.isid.es/Spanish/product/producto_v00.htm.
- WordPress. 2010.** Definicion.de. *Definicion.de.* [Online] 2010. [Cited: 02 17, 2012.] <http://definicion.de/html/>.
- Yanover, David. 2005.** MASTERMAGAZINE. *MASTERMAGAZINE.* [Online] 2 25, 2005. [Cited: 12 14, 2010.] <http://www.mastermagazine.info/termino/6781.php>.
- . 2005.** MASTERMAGAZINE. *MASTERMAGAZINE.* [Online] 02 11, 2005. [Cited: 05 10, 2011.] <http://www.mastermagazine.info/termino/4316.php>.
- Zaninotto, Fabien Potencier y François. 2008.** Symfony: La guía definitiva 1.2. [Online] 2008. [Cited: 6 6, 2012.] www.librosweb.es/symfony/.

GLOSARIO

Patrón: Disciplina de resolución de problemas reciente en la ingeniería del software que ha emergido en mayor medida de la comunidad de orientación a objetos, aunque pueden ser aplicados en cualquier ámbito de la informática y las ciencias en general.

Nodos: Se le llama nodo en la ciencia y otras disciplinas al punto real o abstracto en donde se reúnen las distintas partes de una conexión para comunicarse entre sí. Un nodo, en informática, es un componente que forma parte de una red. En Internet, cada servidor constituye un nodo. Las computadoras que forman parte de una red también son nodos.

El **Instituto Cubano de Radio y Televisión** o **ICRT** es un organismo perteneciente al Estado cubano y de carácter público que se encarga del control de las emisoras de radio y televisión. Fue fundado en 1962. Actualmente es miembro de la Organización de Televisión Iberoamericana y miembro asociado de la Unión Europea de Radiodifusión.

SMTP: Simple Mail Transfer Protocol (SMTP) Protocolo Simple de Transferencia de Correo, es un protocolo de la capa de aplicación. Protocolo de red basado en textos utilizados para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos (PDA's, teléfonos móviles, etc.). Está definido en el RFC 2821 y es un estándar oficial de Internet.