



**Universidad de las Ciencias Informáticas**  
**Facultad 9**

**Título:** Ampliación del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria.

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMATICAS**

**Autor:** Adrian Alberto Alcober Alcover.

**Tutor:** Ing. Eridniel Suárez Contreras.

**Co-Tutor:** Ing. Daniel Sampedro Bello.

## FRASE

---



Los que se enamoran de la práctica sin la teoría son como los pilotos sin timón ni brújula, que nunca podrán saber a dónde van.

**Leonardo Da Vinci**

# DEDICATORIA

---

A mi madre y a mi abuela, por haber sido mi inspiración y la razón de todos mis esfuerzos, por lograr este sueño, que también es de ellas.

# AGRADECIMIENTOS

---

Le agradezco eternamente a mi mamá que durante estos cinco años y toda la vida me han brindado su apoyo, cariño y amor para que un día como hoy pueda estar en este lugar, nada de lo que he logrado hasta hoy hubiera sido posible sin ella, gracias por existir y por ser parte de mi vida, te quiero.

Le agradezco especialmente a mi compañera, amiga y novia Yadira por ser mi sostén en los momentos difíciles, por tener siempre su apoyo y consideración, por ser la persona que es, te quiero así como eres, nunca cambies.

A la familia de Hilda y Deisi, mis otras dos madres, por haberme ayudado tanto o quizás demasiado durante estos 5 años.

A la familia de mi novia que me supo acoger como un miembro más.

A los compañeros de trabajo de mi mamá que me han ayudado tanto.

A Daniel, por todo. Porque sin él no hubiera sido posible este trabajo, por darme las fuerzas y ayudarme a ser mejor profesional, gracias mi hermano.

A los muchachos del proyecto y en especial a Yoendry y Luis Orlando.

A mi tutor y mi oponente por su paciencia, atención y ayuda.

A Yasmany, mi amigo y hermano, por estar ahí conmigo cada uno de los días de estos 5 años en la buenas y en las malas.

A mis amigos, Julio (El Chino), Jorgito, Carlos, Jean, Gilberto, Yanoisis, Reinier Q., José E., Bryan, Humberto, Sergio y muchos más que me tendrán que disculpar por no mencionarlos pero ellos saben quiénes son.

A todos lo amistades de mi novia que de una manera u otra me han ayudado durante todo este tiempo.

A todos, gracias!

# DECLARACION DE AUDITORÍA

---

Declaro que soy el único autor del presente trabajo de diploma y autorizo a la Facultad 9 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente declaración a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año 2010.

Autor:

Tutor:

---

Adrian A. Alcober Alcover

---

Ing. Eridniel Suárez Contreras

# DATOS DE CONTACTO

---

**Tutor:**

**Nombre y apellidos:** Eridniel Suárez Contreras

**Sexo:** M

**Institución:** Universidad de las Ciencias Informáticas (UCI)

**Dirección de la institución:** Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

**Correo electrónico:** [esuarez@uci.cu](mailto:esuarez@uci.cu)

**Teléfono del trabajo:** 07-835-8284

**Categoría docente:** Adiestrado

**Cargo del trabajador:** Profesor

**Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas

**Año de graduación:** 2008

**Institución donde se graduó:** Universidad de las Ciencias Informáticas

**Co-Tutor:**

**Nombre y apellidos:** Daniel Sampedro Bello

**Sexo:** M

**Institución:** Universidad de las Ciencias Informáticas (UCI)

**Dirección de la institución:** Carretera a San Antonio de los Baños, km 2 ½, Boyeros, Ciudad de La Habana.

**Correo electrónico:** [dsampedro@uci.cu](mailto:dsampedro@uci.cu)

**Categoría docente:** Adiestrado

**Cargo del trabajador:** Profesor

**Título de la especialidad de graduado:** Ingeniero en Ciencias Informáticas

**Año de graduación:** 2009

**Institución donde se graduó:** Universidad de las Ciencias Informáticas

# OPINIONES Y AVALES

---

El Trabajo de Diploma, titulado Ampliación del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria, fue realizado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
- Parcialmente en un \_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes (cuantificar):

---

---

---

---

---

---

---

---

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a <valor en MN o USD del efecto económico>

Y para que así conste, se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2010.

\_\_\_\_\_  
Representante de la entidad

\_\_\_\_\_  
Cargo

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Cuño

# RESUMEN

---

A nivel mundial los medios de comunicación juegan un papel importante en cuanto a la divulgación de la información, y en el caso de que se utilizaran para incentivar el conocimiento sano de las personas entonces se logra una televisión y una radio comprometida con el desarrollo intelectual. Ese es el propósito que persigue la Universidad de las Ciencias Informáticas con la creación de la Dirección de Televisión Universitaria (DTU), la que se encarga de apoyar al sistema educacional de la institución con la grabación y publicación de las teleconferencias para los distintos años y cursos de posgrado.

Actualmente la DTU contiene un conjunto de procesos que son implícitos pero de suma importancia para lograr su objetivo principal, estos procesos se encuentran automatizados en un Sistema de Gestión (SGPDTU) que centraliza y controla el flujo de información manejada en la DTU. Aunque este sistema de gestión funciona como se esperaba inicialmente, las directivas han considerado que se le deben agregar otras funcionalidades que aumenten su desempeño y mejoren su calidad.

En el Trabajo de Diploma se recogerá una investigación, que dará como resultado la implementación de las nuevas funcionalidades que le serán agregadas al SGPDTU, con el objetivo de optimizar el trabajo de la DTU.

## Palabras Claves

Medios de Comunicación, Dirección de Televisión Universitaria, Procesos, Sistema de Gestión.

# ÍNDICE

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO 1</b> .....	5
<b>1.1 INTRODUCCIÓN</b> .....	5
<b>1.2 CONCEPTOS ASOCIADOS EL DOMINIO DEL PROBLEMA</b> .....	6
<b>1.3 TENDENCIAS Y TECNOLOGÍAS ACTUALES PARA EL DESARROLLO DE UN SISTEMA DE GESTIÓN</b> .....	7
1.3.1 MODELO ARQUITECTÓNICO CLIENTE – SERVIDOR.....	7
1.3.2 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	9
1.3.2.1 Rational Unified Process (RUP) .....	10
<b>1.4 HERRAMIENTAS DE DESARROLLO</b> .....	12
1.4.1 LENGUAJES DE PROGRAMACIÓN.....	12
1.4.1.1 Lenguajes del lado del cliente.....	12
1.4.1.2 Lenguajes del lado del servidor.....	17
1.4.2 FRAMEWORK.....	19
1.4.2.1 Symfony.....	19
1.4.2.2 J2EE (Java 2 Enterprise Edition).....	21
1.4.3 ENTORNO DE DESARROLLO INTEGRADO (IDE) .....	23
1.4.3.1 Eclipse.....	23
1.4.3.2 NetBeans .....	24
1.4.4 SELECCIÓN DE LAS HERRAMIENTAS DE DESARROLLO.....	25
<b>1.5 CONCLUSIONES PARCIALES</b> .....	28
<b>CAPÍTULO 2</b> .....	29
<b>2.1 INTRODUCCIÓN</b> .....	29
<b>2.2 DESCRIPCIÓN DE LOS SUBSISTEMAS</b> .....	29
<b>2.3 MODELO DE IMPLEMENTACIÓN</b> .....	31

# ÍNDICE

---

<b>2.4 SUBSISTEMAS DE IMPLEMENTACIÓN .....</b>	<b>33</b>
<b>2.5 DIAGRAMA DE COMPONENTES.....</b>	<b>34</b>
2.5.1 Subsistema de Programación. ....	35
2.5.2 Subsistema de Soporte. ....	37
<b>2.6 ESTÁNDARES DE CODIFICACIÓN.....</b>	<b>38</b>
<b>2.7 CONCLUSIONES PARCIALES .....</b>	<b>41</b>
<b>CAPÍTULO 3 .....</b>	<b>42</b>
<b>3.1 INTRODUCCIÓN.....</b>	<b>42</b>
<b>3.2 MÉTODOS DE PRUEBA .....</b>	<b>42</b>
3.2.1 Prueba de Caja Blanca .....	43
3.2.2 Prueba de Caja Negra.....	43
<b>3.3 NIVELES DE PRUEBAS .....</b>	<b>44</b>
<b>3.4 TIPOS DE PRUEBAS.....</b>	<b>45</b>
<b>3.5 DISEÑO DE LAS PRUEBAS.....</b>	<b>48</b>
3.5.1 Casos de pruebas para el Módulo Gestionar Equipamiento. ....	49
3.5.2 Casos de pruebas para el Módulo Gestión de la Programación. ....	51
<b>3.6 CONCLUSIONES PARCIALES .....</b>	<b>55</b>
<b>CONCLUSIONES GENERALES .....</b>	<b>56</b>
<b>RECOMENDACIONES .....</b>	<b>57</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>58</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>60</b>
<b>BIBLIOGRAFÍA.....</b>	<b>62</b>
<b>ANEXOS.....</b>	<b>¡ERROR! MARCADOR NO DEFINIDO.</b>

# ÍNDICE

---

## Índice de Figuras

<b>FIGURA 1:</b> MODELO ARQUITECTÓNICO CLIENTE-SERVIDOR. ....	8
<b>FIGURA 2:</b> FASES Y FLUJOS DE TRABAJOS SEGÚN RUP. ....	11
<b>FIGURA 3:</b> MODELO AJAX. ....	15
<b>FIGURA 4:</b> MODELO DE IMPLEMENTACIÓN. ....	32
<b>FIGURA 5:</b> SUBSISTEMA DE PROGRAMACIÓN. ....	33
<b>FIGURA 6:</b> SUBSISTEMA DE SOPORTE. ....	33
<b>FIGURA 7:</b> DIAGRAMA DE COMPONENTES DEL MÓDULO GESTIONAR PROGRAMACIÓN. ....	35
<b>FIGURA 8:</b> DIAGRAMA DE COMPONENTES DEL MÓDULO GESTIONAR EQUIPAMIENTO. ....	37

## Índice de Tablas

<b>TABLA 1</b> TIPOS DE PRUEBAS QUE SE APLICAN AL SOFTWARE.....	46
<b>TABLA 2</b> CU_ ADICIONAR EQUIPAMIENTO.....	49
<b>TABLA 3</b> CU_ MODIFICAR EQUIPAMIENTO.....	50
<b>TABLA 4</b> CU_ ELIMINAR EQUIPAMIENTO.....	50
<b>TABLA 5</b> CU_ BUSCAR EQUIPAMIENTO.....	51
<b>TABLA 6</b> CU_ ADICIONAR PLAN DE PROGRAMACIÓN.....	51
<b>TABLA 7</b> CU_ MODIFICAR PLAN DE PROGRAMACIÓN.....	52
<b>TABLA 8</b> CU_ ELIMINAR PLAN DE PROGRAMACIÓN.....	53
<b>TABLA 9</b> CU_ VISUALIZAR PLAN DE PROGRAMACIÓN.....	53

# INTRODUCCIÓN

Las tendencias actuales a las que se exponen las Tecnologías de la Información y las Comunicaciones se ven enfocadas a controlar las disímiles vías de información, sobre todo si los datos que se transmiten por estas vías están fuertemente relacionados a sectores económicos y sociales. Los medios de comunicación se han convertido en el motor principal que promueve la información con el fin de divulgarla, siendo la radio y la televisión los medios que cuentan con más auge en la actualidad.

En la Universidad de las Ciencias Informáticas se encuentra la Dirección de Televisión Universitaria (DTU), que funciona como una institución que se encarga de utilizar la televisión no solo como medio de entretenimiento, sino con fines educativos.

*Esta Dirección fue creada en los mismos inicios de la Universidad debido a la necesidad de un sistema de comunicación que abarcara el campus universitario y llevara a ellos en principio un conjunto de informaciones importantes para el desarrollo de la vida y la convivencia dentro de la Institución; inmediatamente esta idea fue desarrollándose y abarcando otras ramas de suma importancia como la docencia (3). Las grabaciones de teleclases y conferencias impartidas que posteriormente son expuestas al uso de la comunidad universitaria, infieren entre las funciones principales de dicho departamento, aunque también se le da cobertura a los disímiles eventos extracurriculares de la universidad, como son: juegos deportivos, festivales de artistas aficionados, eventos científicos, procesos políticos, así como conciertos y visitas de importantes personalidades de la ciencia y la cultura nacional e internacional. La DTU desarrolló además un canal interno, que brinda espacio a la radio universitaria a través de su emisora Radio Ciudad Digital (RCD).*

La planificación del trabajo que se realiza en la DTU incluye varios procesos, que interrelacionados conforman el plan de producción de dicho departamento; dígame pedidos y gestión de medias, distribución del trabajo, planificación de la programación semanal a transmitir y administración de cada uno de los medios técnicos con que cuenta la DTU. Cada uno de esos procesos está dirigido a trabajadores y estudiantes que según el rol que desempeñan, se les asignan los recursos que sean

# INTRODUCCIÓN

---

necesarios para una actividad determinada. Estos procesos generan gran cantidad de información y por el alcance que contienen han de ser gestionados de la mejor manera. Un flujo de datos tan diverso es difícil de monitorear aunque se encuentren manipulados por un sistema que los centraliza, por lo que se hace necesario incorporarle otras funcionalidades que apoyen su ejecución dinámica.

Actualmente existe un sistema automatizado que permite conformar el plan de producción, al igual que el procesamiento de las solicitudes de los medios por parte de los estudiantes y trabajadores vinculados a la DTU. Aunque el SGPDTU cumple con sus propósitos fundamentales, aún se necesita incorporarle otras funcionalidades, que integradas apoyen la calidad de sus servicios y a la toma de decisiones, por lo que el **problema a resolver** para esta investigación figura en la *inexistencia de un conjunto de funcionalidades que han de ser integradas al sistema de gestión automática para la asignación de recursos correspondientes al plan de producción de la Dirección de Televisión Universitaria en la UCI.*

Donde *implementar un conjunto de funcionalidades que serán integradas a la aplicación web que permite automatizar los procesos que realiza la Dirección de Televisión Universitaria de acuerdo con la asignación de recursos, que conformen el plan de producción;* será el **objetivo general** que permitirá resolver la problemática existente.

Se hace necesario centrar la investigación en *los procesos de la asignación de recursos correspondientes al plan de producción de la DTU*, lo que constituye el **objeto de estudio**, que dará cumplimiento al Objetivo General planteado con anterioridad. Enmarcando la investigación en *la informatización de los procesos para la asignación de recursos correspondientes al plan de producción de la DTU*, lo que conforma el **campo de acción**; y así se da cumplimiento a la **idea a defender** de que *con la implementación de funcionalidades que serán integradas al SGPDTU, se facilitará una correcta toma de decisiones en pos de conformar el plan de producción de la DTU.*

En aras de darle cumplimiento al objetivo propuesto se plantean las siguientes **tareas de la investigación científica**:

- Seleccionar las distintas tendencias y tecnologías para el desarrollo de software que existen a nivel internacional, nacional y de la Universidad.

# INTRODUCCIÓN

---

- Identificar las distintas las tendencias y tecnologías actuales, así como las herramientas de desarrollo que la soportan.
- Valorar las distintas tendencias y tecnologías actuales, así como de las herramientas de desarrollo que la soportan.
- Construir el Modelo de Implementación.
- Implementar el sistema diseñado.
- Integrar con otros componentes o partes del sistema.
- Desarrollar pruebas de funcionalidad a los componentes que certifiquen cada una de las funcionalidades implementadas.

La actual investigación posee como posibles resultados la ampliación del Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria, la elaboración del diagrama de componentes de los subsistemas, así como del modelo de Implementación. Contempla además, el documento de estándares de codificación, el documento de defectos encontrados, y el prototipo funcional de los subsistemas; que solo serán logrados con el cumplimiento de las tareas propuestas anteriormente.

Los Métodos de la Investigación Científica permiten desarrollar con calidad y exclusividad todo el proceso investigativo de este trabajo de diploma, el cual se ha apoyado tanto en los teóricos como en los empíricos ejemplificados a continuación:

## **Métodos Teóricos:**

El método de **análisis - síntesis**, permite analizar los procesos relacionados con la gestión del plan de producción, elaborar un resumen para concluir que es lo que realmente se necesita ampliar.

A través del **histórico - lógico** se investiga toda la documentación del sistema existente en pos de optimizar el desarrollo de las nuevas funcionalidades.

La **modelación** permite diseñar los diagramas y componentes en correspondencia con los procesos ingenieriles del trabajo de diploma.

# INTRODUCCIÓN

---

El presente Trabajo de Diploma está estructurado en 3 capítulos, donde el **Capítulo 1** registra la fundamentación teórica del diseño metodológico expuesto en la introducción, se describe cada una de las herramientas de desarrollo a utilizar así como las tendencias y tecnologías actuales para el desarrollo de un sistema de gestión.

El **Capítulo 2** ilustra mediante diagramas la dinámica de la aplicación a través el uso del lenguaje UML, también se definen y ejemplifican los artefactos generados por el rol de implementador.

En el **Capítulo 3** se valida la solución propuesta, atendiendo a la calidad del software se aplicaron a algunas de las funcionalidades del sistema pruebas para garantizar la calidad, en aras de un mejor aprovechamiento.

# Capítulo 1

## Fundamentación Teórica

### 1.1 Introducción

Actualmente una de las aplicaciones de la informática que se explota en todos los campos y sectores sociales y económicos son los Sistemas de Gestión, los cuales se han convertido en potenciales software que facilitan la toma de decisiones. El SGPTDU es uno de estos sistemas, que de manera particular monitorea las actividades de la Dirección de Televisión Universitaria (DTU) de la Universidad de Ciencias Informáticas (UCI). A pesar de que el SGPTDU es un software que funciona en su forma básica, aún está en crecimiento, pues se considera que debe incorporar un conjunto de funcionalidades que lo hagan más eficiente y confiable, lo cual es el propósito de este Trabajo de Diploma.

El presente capítulo se encarga de brindar información referente a la investigación realizada en cuanto a las técnicas y tecnologías aplicadas en la creación de este tipo de sistemas, enfocándose y respetando siempre las políticas establecidas en la primera versión del producto.

## 1.2 Conceptos asociados el dominio del problema

Antes de abundar en la temática que concierne el presente capítulo, es importante describir algunos conceptos que se destacan en la investigación. La definición de “sistema de gestión” ha sido referenciada en diversas bibliografías y considerando su significación dentro del contexto, se ha decidido dejar constancia de algunos significados propuestos por varios autores, para en un final evidenciar la documentación con un criterio propio enmarcado en la aplicación.

“Es el conjunto de normas interrelacionadas por las cuales se administran de forma ordenada las actividades referentes a una acción.” (4)

“Es la disciplina de organizar y administrar recursos de manera tal que se pueda culminar todo el trabajo requerido dentro del alcance, el tiempo y coste definidos.” (5)

“Conjunto de procesos destinados a asegurar la funcionabilidad de todo producto obtenido durante cualquiera de las etapas del desarrollo de un Sistema de Información, a través del estricto control de los cambios realizados sobre los mismos.” (7)

**Sistema de Gestión:** es una estructura probada para la gestión y mejora continua de los procedimientos y procesos de la organización. A demás ayuda a lograr los objetivos de la organización mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

**SGPDTU:** Es un sistemas de gestión que mejora los procedimientos y optimiza los procesos referentes a la Dirección de Televisión Universitaria; dígase pedidos de media, distribución del trabajo, gestión de medias, gestión del equipamiento técnico y creación de plan de la programación televisiva semanal.

## 1.3 Tendencias y Tecnologías actuales para el desarrollo de un Sistema de Gestión.

Cuando se menciona el tema de las tendencias y tecnologías actuales para el desarrollo de algún Sistema de Gestión, se alude principalmente a que en su mayoría utilizan un entorno Web por los beneficios que este ofrece en cuanto a conexiones múltiples, utilidad, disminución de recursos de la PC y seguridad informática.

*Permite además tener clientes independientes de tecnología y en las estaciones de trabajo. Sólo se requiere tener un navegador Web con el cual se accede al sistema, lo que reduce las necesidades de recursos de las estaciones de trabajo. El capital humano para el mantenimiento del sistema disminuye pues no es necesaria la instalación ni configuración individualizada. Facilita implementar mecanismos de seguridad como la autenticación IP y el monitoreo de los accesos. La administración es de forma centralizada. Permite tener la información almacenada en mismo ordenador, lo cual elimina la necesidad de hacer replicas periódicas y mejor resguardo de la información con copias de respaldo de la información. Mediante sistemas informáticos sobre la Web se obtienen resultados superiores en cuanto al trabajo con información, las intranets son una vía barata y efectiva de proveer información a toda la organización para la cual se implementa. (7)*

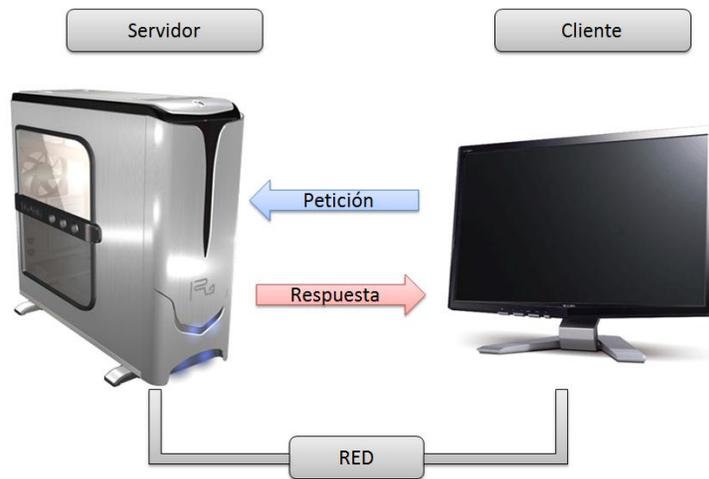
Debido a la autenticidad del planteamiento anterior, el SGPTDU es un sistema realizado sobre un entorno Web y que además se guía utilizando el Modelo Arquitectónico Cliente-Servidor, el cual se argumenta a continuación.

### 1.3.1 Modelo Arquitectónico Cliente – Servidor

El esquema cliente-servidor "es un modelo de computación en el que el procesamiento requerido para ejecutar una aplicación o conjunto de aplicaciones relacionadas se divide entre dos o más procesos que cooperan entre sí " (8). Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y los procesos clientes sólo se ocupan de la interacción con el usuario.

# Fundamentación Teórica

---



**Figura 1: Modelo Arquitectónico Cliente-Servidor.**

Los principales componentes del esquema cliente-servidor son entonces los clientes, los servidores y la infraestructura de comunicaciones.

Los clientes interactúan con el usuario, usualmente en forma gráfica. Frecuentemente se comunican con procesos auxiliares que se encargan de establecer conexión con el servidor, enviar el pedido, recibir la respuesta, manejar las fallas y realizar actividades de sincronización y de seguridad.

*Los servidores proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Por las razones anteriores la tecnología asociada al hardware de los servidores es más poderosa que la de los clientes. Además deben manejar servicios como administración de la red, mensajes, control y administración de la entrada al sistema, contabilidad, auditoría y recuperación.*

(9)

Para que los clientes y los servidores puedan comunicarse se requiere una infraestructura de comunicaciones, la cual proporciona los mecanismos básicos de direccionamiento y transporte. La

# Fundamentación Teórica

---

mayoría de los sistemas Cliente-Servidor actuales se basan en redes locales y por lo tanto utilizan protocolos no orientados a conexión, lo cual implica que las aplicaciones deben hacer las verificaciones. La red debe tener características adecuadas de desempeño, confiabilidad, transparencia y administración.

Con la utilización del Modelo Cliente-Servidor se reduce el costo del hardware requerido y de producción de software, llevando las aplicaciones a plataformas más baratas, así como la disminución del tiempo de ejecución. Además el esquema Cliente-Servidor es que facilita el suministro de información a los usuarios; esto es así porque, por un lado proporciona una mayor consistencia a la información al contar con un control centralizado de los elementos compartidos, y por otro, porque facilita la construcción de interfaces gráficas interactivas, las cuales pueden hacer que los "datos" se conviertan en "información".

Otra de sus ventajas es que también favorece la adaptación a cambios en la tecnología pues facilita la migración de las aplicaciones a otras plataformas y, al aislar claramente las diferentes funciones de una aplicación hace más fácil incorporar nuevas tecnologías en esta.

## 1.3.2 Metodología de desarrollo de software

Para llevar a cabo un proceso de desarrollo de software se hace necesario contar con una guía metodológica que muestre pasos y diagramas con los cuales las actividades realizadas queden debidamente documentadas. *“Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, etc.). Adicionalmente una metodología debería definir con precisión los artefactos, roles y actividades involucrados, junto con prácticas y técnicas recomendadas, guías de adaptación de la metodología al proyecto, guías para uso de herramientas de apoyo, etc. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a una (o algunas) actividades del proceso de desarrollo, por ejemplo, suele hablarse de métodos de análisis y/o diseño”.* (10)

Existen un sin número de Metodologías orientadas y aplicables a las disímiles tendencias de Desarrollo de Software, entre las cuales se encuentra el Proceso Unificado de Desarrollo o Rational Unified Process (RUP), la cual se describe a continuación.

## 1.3.2.1 Rational Unified Process (RUP)

*Es un proceso de desarrollo de software el cual utiliza el lenguaje unificado de modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos (11).*

Es importante destacar que RUP no es una metodología en sí, sino una fusión de varias metodologías que se integran y adaptan a las necesidades de cada grupo de trabajo, guiando su desarrollo a través de pasos claves enfocados en principios y fases ya establecidas. También registra y organiza la producción y mantenimiento de los modelos del sistema. Constituye una guía de cómo usar eficazmente el UML y permite medir la eficiencia de cada proyecto.

Entre sus características se puede apreciar que es guiado y manejado por casos de uso, está centrado en la arquitectura del sistema en construcción, es iterativo e incremental, su desarrollo es basado en componentes y utiliza un único lenguaje de modelado UML, lo cual hace de RUP un proceso integrador.

*La estructura estática del proceso unificado se define en base a cuatro elementos, que son: los Roles (antes workers), que responde a la pregunta ¿quién?, las actividades (activities), que responden a la pregunta ¿cómo?, los productos (artifacts), que responden a la pregunta ¿qué?, y los flujos de trabajo (workflows), que responden a la pregunta ¿cuándo? (12)*

RUP plantea 5 principios:

1. El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico, aunque se debe tener en cuenta el alcance del proyecto.
2. Debe encontrarse un balance que satisfaga los deseos de todos.
3. Los proyectos se entregan en etapas iteradas. En cada iteración se analiza la opinión, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados.

# Fundamentación Teórica

4. Un alto nivel de abstracción previene a los ingenieros de software es directamente de los requisitos a la codificación de software a la medida del cliente. También permite discusiones sobre diversos niveles arquitectónicos. Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes de cuarta generación (SQL, lenguajes de consulta), o esquemas (frameworks).
5. El control de calidad debe en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

El RUP divide el proceso de desarrollo en flujos de trabajo, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases, donde se debe tomar una decisión importante:

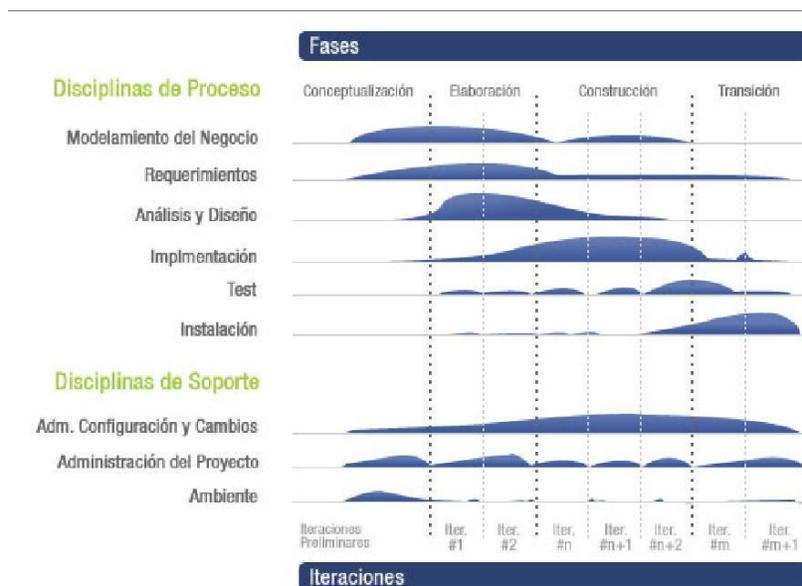


Figura 2: Fases y Flujos de Trabajos según RUP.

Las fases de RUP permiten establecer la oportunidad y alcance del producto, también identifica las entidades externas y los actores con los que trata relacionándolos a los casos de uso.

Con utilizar RUP queda garantizada la planificación, desarrollo y mantenimiento de cualquier sistema, principalmente uno encargado de la gestión de datos o información.

## 1.4 Herramientas de Desarrollo

Todo sistema automatizado necesita para su implementación un conjunto de herramientas en las cuales apoyar su desarrollo. A continuación se presenta un análisis de varias herramientas las cuales fueron agrupadas en: lenguajes de programación, plataformas o frameworks y los entornos de desarrollo integrados (IDE)

### 1.4.1 Lenguajes de Programación

Para la construcción de un sistema automatizado, la selección de los lenguajes de programación ha de ser una actividad fundamental, pues es lo que garantiza la integralidad del futuro software. El SGPDTU se concebirá relacionando un conjunto de lenguajes de programación para la Web, dividido en dos partes, los del lado del cliente y los del lado del servidor, persiguiendo el objetivo de agilizar la actividad monitoreada por el propio sistema.

Los lenguajes del lado del cliente son utilizados con fines de validación de datos y para tratamiento de la interfaz del sistema, garantizando así que la información enviada al servidor sea la correcta y que los usuarios trabajen en una interfaz cómoda y amigable. Los lenguajes del lado del servidor se encargan del envío dinámico de la información en forma de datos al servidor a través de una vía segura, garantizando la calidad del envío y a su vez la rapidez de respuesta por parte del servidor.

#### 1.4.1.1 Lenguajes del lado del cliente

##### Java Script

JavaScript es un lenguaje de interpretado con el que pueden escribirse scripts que funcionarán en el entorno de una página web, descifrado por un explorador. Es importante señalar las atribuciones que posee este lenguaje, que se encarga de la presentación de algún documento, es decir, maneja funciones en torno personalizar la interfaz del sistema en desarrollo. Con el Java Script se consigue la validación de la información como datos, controla el movimiento de imágenes, menús interactivos, animaciones de elementos, entre otros. Además, es soportado por todos los navegadores, es sencillo de aprender y está contemplado por los distintos organismos de normalización.

# Fundamentación Teórica

---

*JavaScript comparte muchos elementos con otros lenguajes de alto nivel. Hay que tener en cuenta que este lenguaje es muy semejante a otros como C, Java o PHP, tanto en su formato como en su sintaxis, aunque por supuesto tiene sus propias características definitorias. Permite diferenciar mayúsculas y minúsculas, por lo que si se escribe alguna expresión en minúsculas, se debe mantener esa expresión en minúsculas a lo largo de todo el programa. También puede encerrar las expresiones que se escriban con una serie de caracteres especiales; estos caracteres se denominan operadores y sirven tanto para encerrar expresiones como para realizar trabajos con ellas. (19)*

JavaScript se caracteriza además, por ser un lenguaje seguro y fiable que permite el filtrado de información. Posee capacidades limitadas, por razones de seguridad, por lo cual no es posible hacer todo con JavaScript, sino que es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros. El código JavaScript se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido. Las características antes mencionadas aseguran las ventajas que obtiene este lenguaje, aunque vale destacar que entre sus principales desventajas se destacan que su código es visible y puede ser leído por cualquiera, además de que el mismo código debe descargarse completamente antes de poderse ejecutar.

## **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. (14)*

Entre sus características principales se encuentran la de manejar las incompatibilidades entre navegadores, pues permite soportar cambios de URLs para luego regresar a ellas, lo que hace que Dojo se convierta en soporte de backward, forward y bookmarks. Otra de sus características es la de ocultar el manejo del XMLHttpRequest, además proporciona un sistema orientado a aspectos, brindando así una amplia gama de opciones concentradas en una sola biblioteca JavaScript y es altamente compatible con navegadores antiguos o versiones anteriores. Dojo representa una ventaja

# Fundamentación Teórica

---

en tema de soporte para Ajax, porque posee la habilidad de degradar cuando Ajax/JavaScript no es completamente soportado por el lado del cliente.

Dojo se provee de una arquitectura singular que utiliza una capa de abstracción con la que se pueden utilizar en un navegador otros transportes o IFrames ocultos además de diferentes formatos de datos. En su diseño se encuentra un sistema de paquetes que facilitan el desarrollo modular. Se inicializa con una jerarquía de paquetes de espacios empezando por un paquete "dojo" y luego se puede cargar cualquier otro paquete vía XMLHttpRequest utilizando las funcionalidades ofrecidas en el arranque.

La importancia y utilidad de Dojo es la de abstraer al desarrollador de las complejidades del DHTML facilitando su entendimiento, implementación y el poder de resolver asuntos relacionados con la navegación.

## **Ajax**

AJAX integra un conjunto de tecnologías obteniendo los mejores méritos de cada una de ellas, uniéndolos en poderosas nuevas formas. Ajax 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 XMLHttpRequest; además de la integración a todos sus atributos de JavaScript.

# Fundamentación Teórica

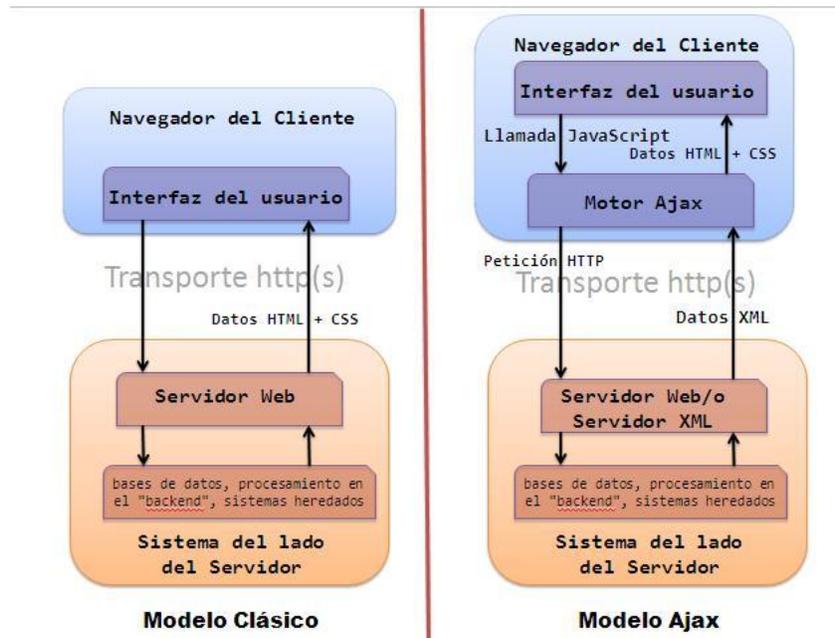


Figura 3: Modelo Ajax.

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 en JavaScript se hace responsable de renderizar la interfaz de usuario y de comunicarse con el servidor en nombre del usuario (15).

Con esta nueva característica que se convierte en una tendencia novedosa en cuanto a la programación de aplicaciones Web, el Motor Ajax garantiza que la interacción del usuario con la aplicación sea independientemente de la comunicación con el servidor. Se obtiene como privilegio que el usuario nunca esté observando una ventana en blanco en su navegador mientras que el servidor está respondiendo a su petición. Si el motor necesita algo del servidor para responder a dicha petición, entonces hace estos pedidos asíncronamente, usualmente usando XML, sin frenar así la interacción del usuario con la aplicación.

# Fundamentación Teórica

---

## HTML

El HTML (HyperText Markup Language) es el lenguaje en el que se escriben los hipertextos del World-Wide Web. Cumple la norma SGML, y permite añadir a un documento de texto:

- La especificación de estructuras del texto. Por ejemplo, títulos, encabezamientos, límites de los párrafos, listas de elementos.
- Estilos: texto enfatizado, citas, entre otros.
- Objetos multimedia: imágenes o sonido.
- Conexiones hipertextuales a otros objetos de la red.

*HTML es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica cómo desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también le indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos ya sea en su computadora o en otras- así como otros recursos de Internet, como FTP y Gopher. (16)*

## Visual Basic Script

*Un lenguaje script de Microsoft derivado de su paquete de programación orientada hacia el objeto. Es la última incorporación a la familia de lenguajes "Basic" que ya comprende a VBA y, sobretodo, a Visual Basic. Permite extender los límites del lenguaje HTML, enlazando los scripts a las páginas Web, para que respondan a los eventos provocados por el usuario ejecutando un código.*

*Microsoft Visual Basic Scripting Edition lleva la ejecución de secuencias de comandos a una variedad de entornos, incluida la ejecución de secuencias de clientes Web en Microsoft Internet Explorer y la ejecución de secuencias de servidores Web en Servicios de Microsoft Internet Information Server(IIS) (17).*

## 1.4.1.2 Lenguajes del lado del servidor

### Personal Home Page (PHP)

"**PHP HyperText Pre-processor**", es un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web soportado por HTML. La sintaxis está heredada de C, Java y Perl. Este lenguaje está orientado para los constructores de páginas Web, permitiéndoles crear páginas dinámicamente generadas de forma rápida.

Lo que hace diferente a PHP es que el código que se interpreta se ejecuta siempre en el servidor. Así, al analizar el script anterior, el cliente recibirá sólo los resultados de la ejecución por lo que es imposible para el cliente acceder al código que generó la página, lo cual se considera una de sus principales ventajas.

*PHP puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente alguno más. Soporta la mayoría de servidores web de hoy en día, incluyendo Apache y Microsoft Internet Information Server. Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos (18).*

PHP posee la capacidad de conectarse con la mayoría de los manejadores de Base de Datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras; además de expandir su potencial utilizando módulos, incluye gran cantidad de funciones implementadas por defecto que agilizan su interpretación y no requiere definición de tipos de variables ni manejo detallado del bajo nivel. PHP constituye un lenguaje libre, por lo que se presenta como una alternativa de fácil acceso.

PHP se caracteriza por tener la posibilidad de ejecutar comandos y abrir conexiones de red en el servidor, se le considera también un lenguaje multiplataforma por lo que su aceptación en la comunidad científica mundial ha sido de forma masiva.

## **JAVA**

Java constituye un lenguaje clasificado dentro de la POO (Programación Orientada a Objetos) desarrollado por Sun Microsystems. Su sintaxis se complementa adquiriendo mucho de C y C++, aunque su modelo de objetos es más simple y elimina las herramientas de bajo nivel que provocan muchos errores, como la manipulación directa de punteros.

*Cualquier aplicación que se desarrolle se apoya en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java). Java incorpora en el propio lenguaje muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores (threads (hilos de procesamiento), ejecución remota, componentes, seguridad, acceso a bases de datos, etc.). Por eso muchos expertos opinan que Java es el lenguaje ideal para aprender la informática moderna, porque incorpora todos estos conceptos de un modo estándar, mucho más sencillo y claro que con las citadas extensiones de otros lenguajes. Esto es consecuencia de haber sido diseñado más recientemente y por un único equipo (19).*

## **Active Server Pages (ASP.NET)**

*Es la última versión de la tecnología ASP pero no es una versión aumentada de esta sino que es enteramente una nueva tecnología para el lado del servidor. ASP.NET es la parte principal del Framework .NET de Microsoft. También contiene un nuevo sistema de entradas de controles orientado a objetos, como las cajas de lista programables y los controles de la validación.*

*Incluye algunas cosas nuevas como:*

- *Soporta los mejores lenguajes*
- *Controles programables.*
- *Componentes basados en XML*
- *Autenticación de usuarios con cuentas y roles.*

# Fundamentación Teórica

---

- *Alta escalabilidad*
- *Código compilado*
- *Fácil configuración y despliegue.*
- *Utiliza ADO.Net, soporta C#, C++ y Visual Basic (VB) pero no Visual Basic Script (VBS) (17).*

## **Processing Extraction Report Language (PERL)**

*Es un lenguaje de programación interpretado, muy utilizado para construir aplicaciones CGI para la Web, se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.*

*Es un lenguaje de libre uso, es decir gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Permite efectuar búsquedas de secuencias de caracteres. Perl se utiliza con fuerza en la escritura de procesos CGI (Common Gateway Interface) instalados en un servidor Web, o para el desarrollo de procesos de mantenimiento de las actividades de un servidor. (17)*

## **1.4.2 Framework**

Un framework se define como un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación.

Los frameworks simplifican el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al programador a crear código más legible y más fácil de mantener.

### **1.4.2.1 Symfony**

*Symfony simplifica al máximo el desarrollo de aplicaciones web profesionales con PHP, utilizando las mejores prácticas y los patrones de diseño más importantes; además incorpora muchas de las ideas del RAD (“desarrollo rápido de aplicaciones”) para conseguir que la programación de las*

# Fundamentación Teórica

---

*aplicaciones sea lo más productiva, correcta y divertida posible. Se considera una de las mejores copias para PHP del famoso framework Ruby on Rails y que también ha tomado las mejores ideas de Rails y de muchos otros frameworks, además ha incorporado ideas propias y el resultado es un framework elegante, estable, productivo y muy bien documentado.*

*Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. 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 (20).*

Symfony emplea el tradicional patrón de diseño MVC (modelo-vista-controlador) para separar las distintas partes que forman una aplicación web. El modelo representa la información con la que trabaja la aplicación y se encarga de acceder a los datos.

La vista transforma la información obtenida por el modelo en las páginas web a las que acceden los usuarios. El controlador es el encargado de coordinar todos los demás elementos y transformar las peticiones del usuario en operaciones sobre el modelo y la vista.

En esta última versión estable de Symfony con la cual se desarrollará, se elimina Propel como la herramienta que se utiliza para el mapeo de objetos a base de datos (ORM, object-relational mapping) y se incluye Doctrine, la cual constituye un PHP ORM para PHP 5.2.3 + que no es más que una potente herramienta PHP BDAL (capa de abstracción de bases de datos). Una de las principales ventajas de Doctrine es la habilidad de poder realizar consultas a bases de datos en un lenguaje OO (orientado a objetos) SQL llamado DQL, denominado así por Hibernate's HQL, y además provee a los desarrolladores un poderosa alternativa SQL para propiciar el desarrollo y maximizar la flexibilidad del código.

Symfony es uno de los frameworks PHP más populares entre los usuarios y las empresas, ya que permite que los programadores sean mucho más productivos a la vez que crean código de más calidad y más fácil de mantener. Symfony es maduro, estable, profesional y está muy bien documentado. Está desarrollado completamente con PHP 5. Ha sido probado en numerosos

proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows.

## 1.4.2.2 J2EE (Java 2 Enterprise Edition)

*J2EE ha sido diseñada para aplicaciones distribuidas que son construidas con base en componentes (unidades funcionales de software), los cuales interaccionan entre sí para formar parte de una aplicación J2EE. Un componente de esta plataforma debe formar parte de una aplicación y ser desplegado en un contenedor, o sea en la parte del servidor J2EE que le ofrece al componente ciertos servicios de bajo nivel y de sistema (tales como seguridad, manejo de concurrencia, persistencia y transacciones). Como ves, J2EE no es sólo una tecnología, sino un estándar de desarrollo, construcción y despliegue de aplicaciones. (21)*

J2EE es un entorno abierto para desarrollar y desplegar servicios multicapa donde pequeñas aplicaciones cliente invocan lógica de negocio que se ejecuta en un servidor de aplicaciones. Comprende un conjunto de servicios, protocolos e interfaces de programación. El lenguaje Java, la máquina virtual Java y los componentes JavaBeans son la base de J2EE.

En un modelo multi-capa, la primera capa es el cliente que normalmente es un navegador Web o una aplicación Java solitaria. Este invoca a la lógica del negocio de una o más capas medias que se están ejecutando sobre hardware dedicado, que a su vez acceden a los datos desde el Enterprise Information Service en la tercera capa.

El modelo de aplicación J2EE encapsula las capas de funcionalidad en componentes de tipos específicos. La lógica de negocio está encapsulada en componentes Enterprise JavaBeans (EJB). La interacción con los clientes pueden representarse a través de:

- Páginas web de HTML plano.
- Páginas web con Applets Java.
- El API Java Servlets.

# Fundamentación Teórica

---

- Tecnología JavaServer Pages.
- Aplicaciones Java solitarias.

Los componentes se comunican de forma transparente usando varios estándares, incluyendo HTML, XML, HTTP, SSL, RMI, IIOP, y otros:

J2EE está compuesto de diferentes componentes:

- **Servlets** — un reemplazo eficiente, independiente de la plataforma para los scripts CGI que responden a solicitudes de clientes.
- **JavaServer Pages (JSP)** — un tipo de script del lado del servidor, que puede generar páginas web dinámicamente.
- **Enterprise JavaBeans (EJB)** — control de sesión del lado del servidor, que encapsula la lógica de negocios y abstracción para acceder a datos persistentes.
- **Java Database Connectivity (JDBC)** — un API que describe una librería estándar Java para acceder a fuentes de datos.
- **Transaction Support** — transacciones declarativas para componentes donde las transacciones pueden expandir componentes y procesos.
- **Java Naming and Directory Interface (JNDI)** — una interface abstracta para servicios de búsqueda de uniones de nombres y directorios.
- **Remote Method Invocation (RM/IIOP)** — una tecnología que permite la comunicación entre objetos distribuidos.
- **CORBA Compatible** — CORBA complementa Java proporcionando un marco de trabajo de objetos distribuidos, servicios para soportar ese marco de trabajo e interoperabilidad con otros lenguajes.

## 1.4.3 Entorno de Desarrollo Integrado (IDE)

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

### 1.4.3.1 Eclipse

Eclipse es una herramienta de las que se denominan “entornos integrados de desarrollo” porque incluye en un sólo programa la capacidad de realizar múltiples operaciones integradas sobre proyectos. Constituye una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. Se trata de un IDE que integra todas las herramientas y funciones necesarias para el desarrollo, recogidas además en una atractiva interfaz que lo hace fácil y agradable de usar.

Cuenta con un editor de texto donde se puede ver el contenido del fichero en el que se está trabajando, una lista de tareas, y otros módulos similares. Si bien las funciones de Eclipse son más bien de carácter general, las características del programa se pueden ampliar y mejorar mediante el uso de plugins.

La arquitectura de plugins de Eclipse permite, además de integrar diversos lenguajes sobre un mismo IDE, introducir otras aplicaciones accesorias que pueden resultar útiles durante el proceso de desarrollo como: herramientas UML, editores visuales de interfaces, ayuda en línea para librerías, entre otros.

*Eclipse es una aplicación multiplataforma que dispone de: un editor de texto, resaltado de sintaxis, compilación en tiempo real, pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, wizards para la creación de proyectos, clases, test, componentes, refactorización. Asimismo, a través de "plugins" libremente disponibles es posible añadir: Control de versiones con Subversion, vía Subclipse; Integración con Hibernate, vía Hibernate Tools, con Aptana, para desarrollar con java scripts. (22)*

## 1.4.3.2 NetBeans

*NetBeans 6.8, el primer IDE que ofrece soporte completo para Java™ Platform Enterprise Edition 6 (Java EE 6) y GlassFish Enterprise Server v3, e incluye innovadoras funcionalidades que facilita la disponibilidad de las mejoras de lenguaje y plataforma para los desarrolladores, acelerando el tiempo de desarrollo de aplicaciones.*

*NetBeans IDE 6.8 porta el beneficio de las últimas capacidades de lenguaje de Java EE 6 que simplifican la creación de aplicaciones en Java, incluyendo desarrollo tipo POJOP, más anotaciones y menos configuración en XML; además incluye nuevas funcionalidades como son:*

- *Mayor soporte PHP: amplía el soporte de lenguajes dinámicos con soporte para PHP 5.3 y el Framework Symfony acelera el desarrollo de aplicaciones PHP Web.*
- *Mejor parametrización C/C ++: Parametrizar y afinar aplicaciones C/C++ con el nuevo indicador Microstate Accounting.*
- *JavaFX: Mejor terminación de código, indicaciones y navegación para JavaFX en el editor NetBeans. (23)*

La plataforma NetBeans es una plataforma RCP (Rich Client Platform) que es utilizada para reducir el tiempo de desarrollo al reutilizar la arquitectura modular de NetBeans. La plataforma utiliza Swing, un kit de herramientas UI estándar que permite a las aplicaciones tener un “look and feel” consistente.

NetBeans IDE es un Entorno de Desarrollo Integrado open source gratuito para desarrolladores de software. Ofrece todas las herramientas necesarias para crear aplicaciones de sobremesa profesionales, empresariales, web y móviles con el lenguaje Java, JavaFX, C/C ++ y lenguajes dinámicos como PHP, JavaScript, Groovy y Ruby. NetBeans IDE es fácil de instalar y listo para usar y se puede ejecutar tanto en Windows, Linux, Mac OS X como en Solaris.

## 1.4.4 Selección de las Herramientas de Desarrollo

Las herramientas de desarrollo aparecen reflejadas en este documento agrupadas en Lenguajes de Programación, Plataformas o Frameworks y los IDE. Se ha expuesto una muestra de sus atribuciones y la comparación entre las mismas arrojará el resultado de la selección realizada.

Existe una tendencia a que toda aplicación Web se aprecie con un entorno amigable para sus futuros usuarios por lo que entre las características del Java Script se puede encontrar esta peculiaridad y entre otras tan importantes como la validación de los datos. Otro de los lenguajes seleccionados es el Ajax y es precisamente por ser una tecnología acumulativa de XML y DOM, escogiendo lo mejor de ambas así como del mismo Java Script, lo que proporciona a la Web el poder de brindar una amplia gama de servicios. También se encuentra el Dojo, que en sí no está clasificado como un lenguaje de programación, pero es considerado una de las librerías más prometedoras integradas a Ajax y Java Script. Los lenguajes seleccionados responden al lado del cliente, pero aún el manejo de los datos y su traslado hacia el servidor han de ser tratados por otro lenguaje. PHP brinda más que eso, es simplemente soportado por innumerables servidores de base de datos y es su característica más simbólica, por la cual es el elegido a integrar con Java Script, Ajax, HTML y Dojo.

Con la obtención de los lenguajes a utilizar, se prosigue a seleccionar la plataforma de desarrollo y los parámetros a medir han de ser precisamente el soporte de los lenguajes y la tendencia al libre acceso, es decir a la implantación de las normas del Software libre. De las dos plataformas analizadas, todas soportan los lenguajes escogidos pero solo una cumple con el requisito de las normas del software libre y es el Symfony, que además brinda al desarrollador comodidad al implementar un sistema. Symfony posee características exclusivas y se exponen a continuación:

**Escalable:** Symfony es infinitamente escalable si se disponen de los recursos necesarios.

**Probado:** Symfony ha sido probado con éxito durante varios años en aplicaciones muy diferentes. Desde sitios web con millones de usuarios (del.icio.us, Yahoo Bookmarks, Yahoo Answers) hasta otros miles de sitios pequeños y medianos.

# Fundamentación Teórica

---

**Soporte:** Symfony sigue una política de tipo LTS (long term support). Las versiones estables se mantienen durante 3 años sin cambios pero con una continua corrección de los errores conocidos.

**Licencia:** Symfony utiliza una licencia MIT, con la que puedes hacer aplicaciones web comerciales, gratuitas y/o de software libre.

**Compromiso:** La empresa que ha creado Symfony no vive del framework, sino de las aplicaciones que hace con él. Lo que se traduce que aspectos como el rendimiento, la buena documentación, el soporte muy largo y otros son de igual interés para sus creadores que para los desarrolladores que lo utilizan.

**Código:** Desde su primera versión Symfony ha sido creado para PHP 5.

**Seguro:** Se puede controlar hasta el último acceso a la información e incluye por defecto protección contra ataques XSS y CSRF.

**Documentado:** Se trata del framework PHP mejor documentado, miles de páginas en el wiki oficial, tutoriales de hasta 250 páginas y un libro gratuito de casi 500 páginas que además está completamente traducido al español.

**Calidad:** su código fuente incluye más de 8.000 pruebas unitarias y funcionales.

**Internacionalización:** Se pueden crear aplicaciones en varios idiomas. La internacionalización está integrada en el framework, es muy completa y está probada en aplicaciones reales.

En el tercer grupo de las herramientas se encuentran los IDE y ya habiendo seleccionado los lenguajes y el framework solo se ha de revisar el parámetro de su completa integración con los mismos. Aunque NetBeans y Eclipse presentan similitudes en cuanto a las funcionalidades incluidas, el completamiento del código en el NetBeans ofrece las variables de Symfony adecuadas en los archivos de sus vistas. Además, los proyectos de Symfony existentes son reconocidos y también se pueden crear otros nuevos. Otra de sus ventajas es que posee atajos de teclado asignables para acciones específicas de Symfony; es de fácil navegación entre vistas y acciones, de ida y vuelta y brinda la posibilidad de ejecutar comandos de Symfony. Por lo que su característica fundamental se

# Fundamentación Teórica

---

basa en la integración ciento por ciento con el Framework de desarrollo escogido, a lo que se asume como IDE para el desarrollo.

## 1.5 Conclusiones Parciales

Es de singular apreciación que la aplicación que se desarrollará mantendrá algunos de los estándares definidos por la primera versión del software, dígase frameworks, lenguajes de programación, metodología de la investigación, modelos y patrones arquitectónicos. Por lo que se mantiene el Symfony como framework de desarrollo; el UML integrado al RUP como metodología de la investigación; los lenguajes PHP, JavaScript, HTML, además del AJAX y Dojo los cuales se integran a la lista con vista de mejorar las funcionalidades que le serán añadidas a la aplicación Web; así mismo se mantiene la jerarquía de la Arquitectura Cliente–Servidor y como Estilo Arquitectónico el Modelo Vista – Controlador el cual se mantiene incluido con el Symfony.

Aunque en la versión anterior del producto se utilizaba como IDE de desarrollo el Eclipse, se realizó un análisis de sus características además de su integración al Symfony, se decidió como nuevo IDE el NetBeans, siendo este compatible e integrable al framework escogido.

# Capítulo 2

## Ampliación del Sistema

### 2.1 Introducción

En el presente capítulo se exponen los artefactos generados por el Rol de Implementador según RUP, siendo uno de estos, el Modelo de Implementación en términos de componentes para de cada uno de los subsistemas que se desean ampliar. A demás se definirán los estándares de codificación necesarios que aseguran una mejor calidad y comprensión del código, y se dará una descripción de los módulos implicados en dicha ampliación.

### 2.2 Descripción de los Subsistemas

El Sistema de Gestión de Procesos de la DTU cuenta con uno conjunto de subsistemas que validan su implementación; debido a que el desarrollo del sistema está basado en el framework Symfony y este en su estructura divide las aplicaciones en módulos para una mejor comprensión y desarrollo, cada uno de los subsistemas de la aplicación está dividido en varios módulos. A continuación se describen los subsistemas y módulos correspondientes al SGPDTU:

- Subsistema de Administración: Posibilitará al administrador del sistema (Persona encargada de Administrar y Gestionar los principales permisos de la aplicación) el control y la validación todo aquel personal que accede a la aplicación, además de velar porque cada usuario realice solo lo que le es permitido según el rol que desempeñen.

# Ampliación del Sistema

---

- Módulo de Gestión de Usuarios: Permitirá adicionar, modificar y eliminar cualquier usuario del sistema, así como también podrá buscar un determinado usuario dentro del sistema.
- Módulo de Gestión de Rol: Permitirá adicionar, modificar y eliminar cualquier rol referente a un usuario del sistema, así como también podrá buscar un determinado rol dentro del sistema.
- Subsistema de Gestión de Medias: Posibilitará al especialista de la DTU (Persona encargada de Gestionar todo lo referente al plan de Producción) gestionar y controlar todo lo referentes a las medias que se almacenan en el sistema.
  - Módulo de Gestión de Medias: Permitirá adicionar, modificar y eliminar cualquier media insertada previamente al sistema, así como también permitirá realizar una búsqueda de una determinada media.
  - Módulo de Gestión de Plan de Producción: Permitirá adicionar, modificar y eliminar los planes de producción semanales por el cual se deben de regir los distintos trabajadores pertenecientes a la DTU, y permitirá además crear reportes de dicho plan.

Como complemento a las funcionalidades del SGPDTU se le incorporan 2 nuevos subsistemas que amplían y mejoran la calidad de sus servicios y la toma de decisiones:

- Subsistema de Programación: Posibilitará que el Especialista de la DTU (Persona encargada de Gestionar todo lo referente al plan de Programación) la gestión del plan de la programación que se transmitirá semanalmente.
  - Módulo Gestión de la Programación: Permitirá adicionar, modificar y eliminar cualquier valor referente al plan de producción insertado en el sistema, así como también permitirá visualizar un determinado plan de programación que se transmitirá por los canales televisivos de la Universidad de las Ciencias Informática.
- Subsistema de Soporte: Posibilitará que el Especialista de la DTU gestione de todo lo referente equipamiento técnico en el DTU, dígase el control de las cámaras de televisión, cableado,

# Ampliación del Sistema

---

micrófonos, etc.; pero se centra fundamentalmente la creación de reportes de todos los procesos que se realizan en la DTU.

- Módulo de Gestión de Equipamiento: Permitirá adicionar, modificar y eliminar todo el equipamiento técnico con el cual cuenta la DTU.
- Módulo de Gestión de Reporte: Permitirá crear, mostrar e imprimir reportes de todos los procesos que se desarrollen el sistema; reportes que no solamente estarán conformados por los datos que generen los diferentes módulos de la aplicación, sino que además estarán compuestos por gráficas y tablas comparativas que evaluarán todo lo referente a trabajo que se realiza día a día en la DTU, en aras de llevar un estricto control sobre todo el sistema.

## 2.3 Modelo de Implementación

El modelo de implementación muestra cómo se desarrollan los elementos de un sistema en términos de componentes. En él se describen cómo se organizan los componentes y cómo dependen unos de otros.

En el presente epígrafe se muestra el modelo de implementación correspondiente al SGPDTU que representan componentes que hacen referencia a librerías del propio framework de desarrollo, tal es el caso de Doctrine para el mapeo de objetos a base de datos (ORM, de “object-relational mapping”), el cual se encarga de la generación del modelo, ya que crea una estructura básica de las clases y genera automáticamente el código necesario para la interacción con la base de datos, proporcionando persistencia para los objetos y un fácil trabajo con las consultas.

A continuación se muestra el modelo de implementación del Sistema de Gestión de Procesos para la DTU con los subsistemas que lo componen.

# Ampliación del Sistema

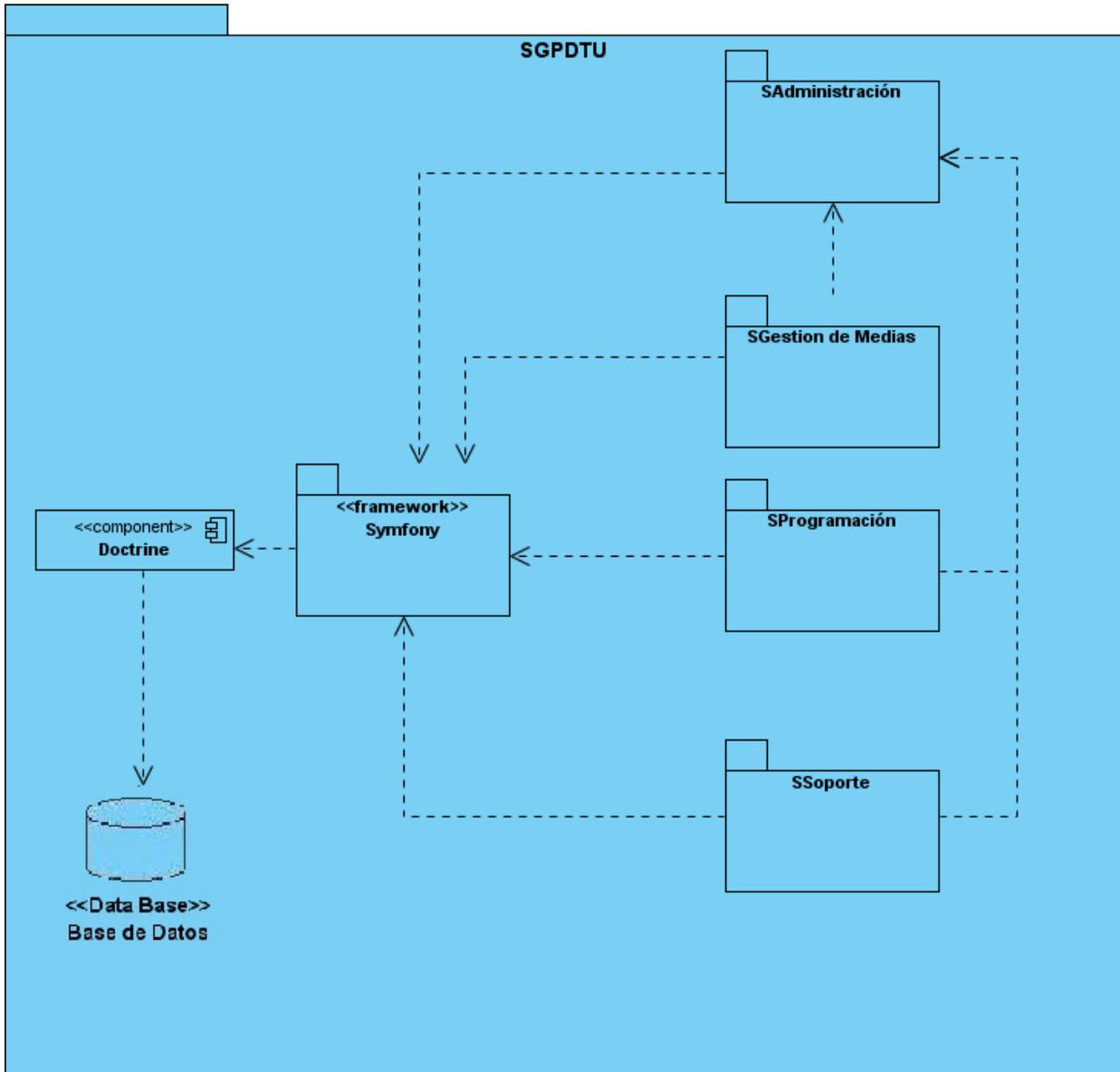
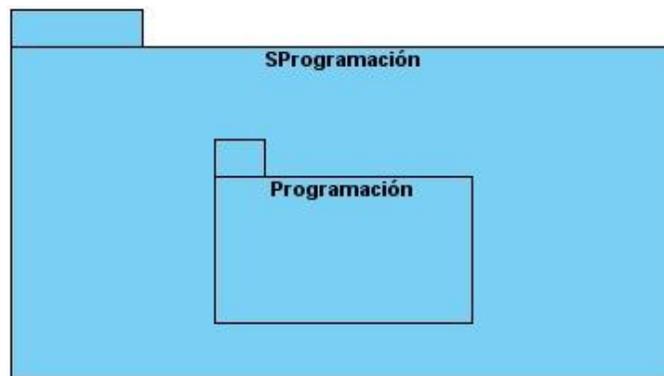


Figura 4: Modelo de Implementación.

## 2.4 Subsistemas de Implementación

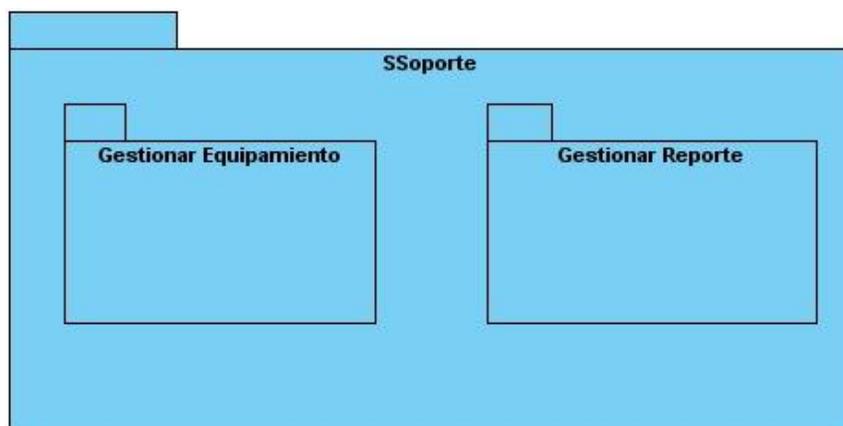
Los subsistemas de implementación constituyen componentes que son utilizados para separar el sistema en pequeños subsistemas que pueden ser integrados y probados de forma separada.

Seguidamente se muestran los subsistemas de implementación que son partícipes de la ampliación del SGPDTU.



**Figura 5: Subsistema de Programación.**

El subsistema de programación está conformado por el módulo de programación el cual constituye el componente o paquete, que a su vez estará compuesto por un conjunto de componentes permitirán manejar todo lo referente a la gestión de la programación en el sistema.



**Figura 6: Subsistema de Soporte.**

# Ampliación del Sistema

---

El subsistema de soporte está conformado por dos módulos, donde el módulo de gestionar equipamiento constituye el componente o paquete, que a su vez estará compuesto por un conjunto de componentes permitirán manejar todo lo referente a la gestión de los equipamientos técnicos en la DTU y además este subsistema cuenta con el subsistema de reporte el cual queda excluido de la ampliación del sistema.

## 2.5 Diagrama de Componentes

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema, muestran y organizan las relaciones de dependencia entre los componentes y subsistemas que lo conforman. La metodología RUP enfatiza en el uso del diagrama de componentes para modelar los subsistemas de implementación. Los componentes representan los elementos de un modelo dentro de un paquete, como son las clases en el modelo de diseño, los cuales se estereotipan de la siguiente:

- <<executable>> programas que se ejecutan en un nodo.
- <<file>> ficheros de datos o código fuente.
- <<library>> librerías estáticas o dinámicas.
- <<table>> tabla de base de datos.
- <<document>> documento.

Seguidamente se muestran los diagramas de componentes relacionados con los subsistemas presentados anteriormente.

## 2.5.1 Subsistema de Programación.

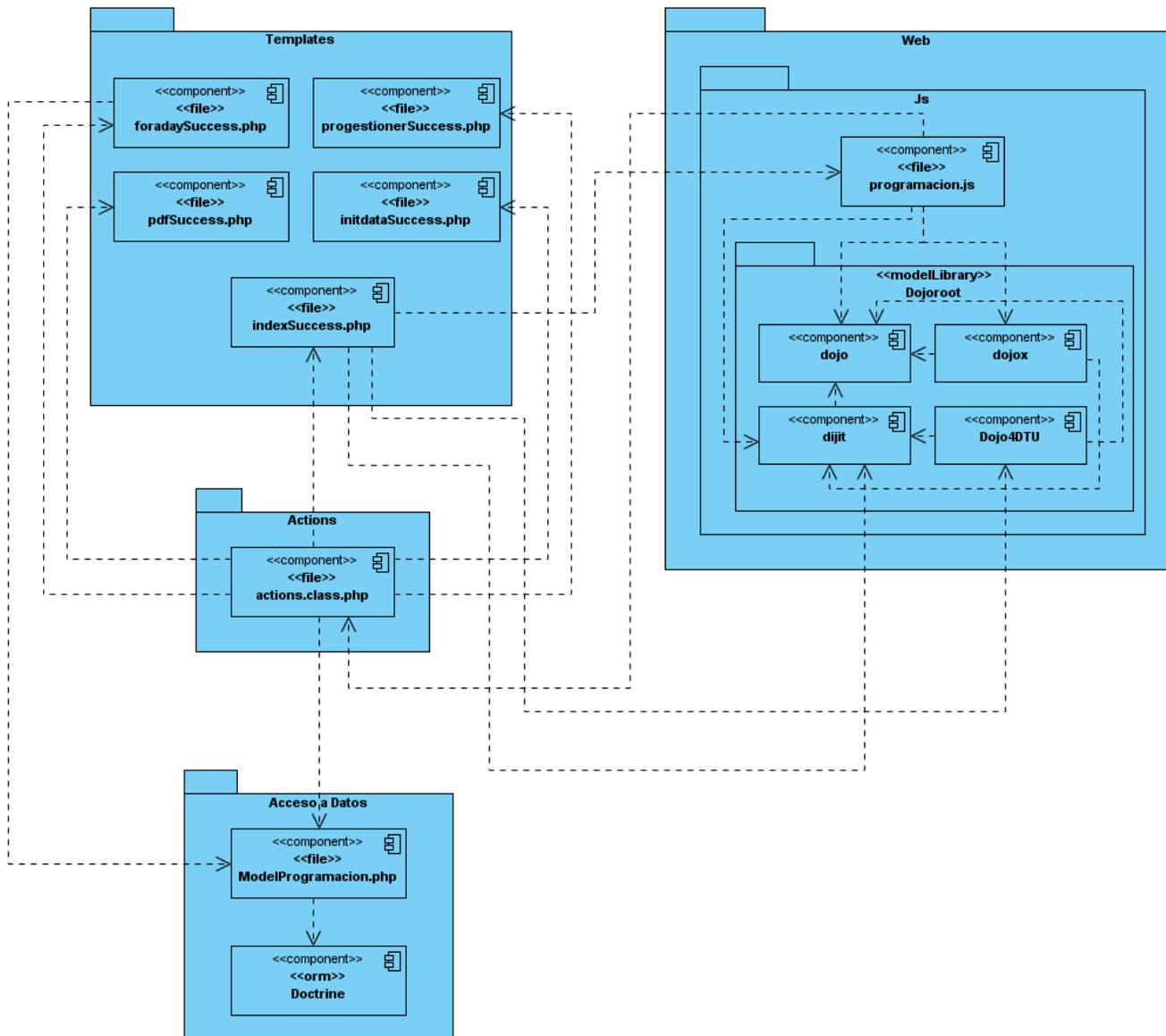


Figura 7: Diagrama de Componentes del Módulo Gestionar Programación.

El módulo gestionar programación está dividido en 4 paquetes de componentes, a través de los cuales se modelan cada una de las clases, ficheros e interacciones necesarias para su implementación.

# Ampliación del Sistema

---

El paquete Acceso a Datos contiene todas las clases referentes a la base de datos, creadas por el ORM Doctrine, las cuales son utilizadas por el paquete Actions, que contienen toda la lógica de la aplicación en cuanto a la gestión de los datos, y que también es el encargado de la declaración de las variables que son usadas en la vista.

El paquete Templates contiene además el fichero indexSuccess.php se encarga de producir las paginas utilizando los paquetes Js y CSS respectivamente, en el paquete Js está presente fichero programación.js el cual constituye el principal elemento del subsistema porque a través de él se describen todos los elementos que se utilizan para la construcción de la interfaz dinámica, tratamiento y validación de los datos, e integración de los lenguajes de programación a través del paquete Dojoroot y del lenguaje programación Ajax.

## 2.5.2 Subsistema de Soporte.

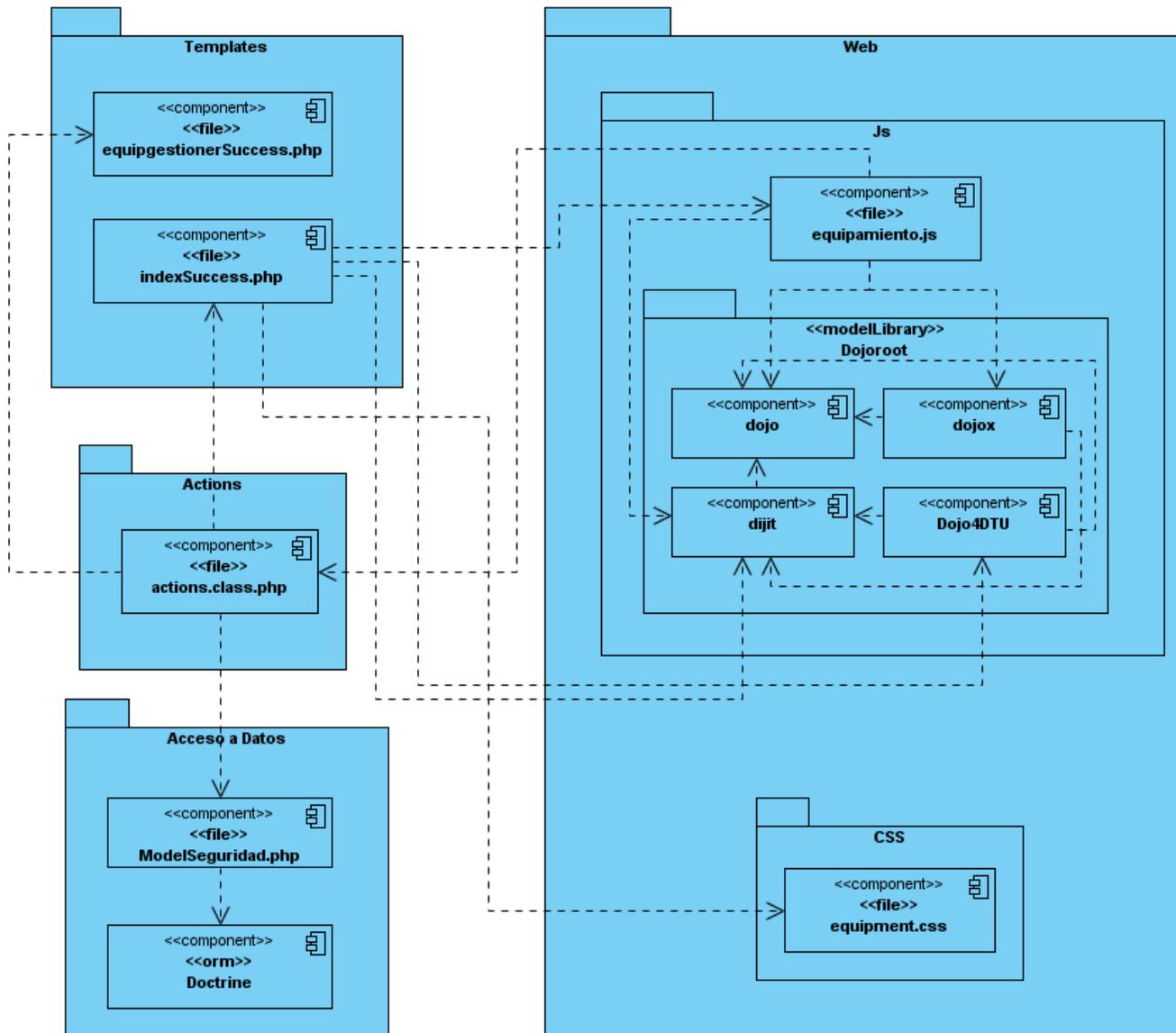


Figura 8: Diagrama de Componentes del Módulo Gestionar Equipamiento.

# Ampliación del Sistema

---

El módulo gestionar equipamiento está dividido en 4 paquetes de componentes, a través de los cuales se modelan cada una de las clases, ficheros e interacciones necesarias para su implementación.

El paquete Acceso a Datos contiene todas las clases referentes a la base de datos, creadas por el ORM Doctrine, las cuales son utilizadas por el paquete Actions, que contienen toda la lógica de la aplicación en cuanto a la gestión de los datos, y que también es el encargado de la declaración de las variables que son usadas en la vista.

El paquete Templates contiene además el fichero indexSuccess.php se encarga de producir las paginas utilizando los paquetes Js y CSS respectivamente, en el paquete Js está presente fichero equipamiento.js el cual constituye el principal elemento del subsistema porque a través de él se describen todos los elementos que se utilizan para la construcción de la interfaz dinámica, tratamiento y validación de los datos, e integración de los lenguajes de programación a través del paquete Dojoroot y del lenguaje programación Ajax.

## 2.6 Estándares de Codificación

Los estándares de codificación constituyen el elemento principal por el cual se rigen los desarrolladores para asegurar que el código sea de alta calidad y que no contenga gran cantidad de errores, permitiendo así que el trabajo en equipo sea más efectivo y que los programadores puedan modificar el código gracias a la facilidad de programación y a la estandarización de la misma.

A continuación se describen una serie de pautas específicas a seguir para el desarrollo con los lenguajes de programación utilizados en la construcción del sistema:

- Saltos de Línea
  - Añadir un salto de línea después del cierre de los paréntesis de los parámetros.
  - Añadir un salto de línea después un punto y coma, cuando termina la sentencia.

# Ampliación del Sistema

---

- Documentar el Código:
  - Las secciones de código más complicadas e inusuales deberán ser documentadas. Cada variable y arreglo tendrán que estar comentados en el lugar que sean definidos para que su función pueda ser comprendida posteriormente.
- Espacios y líneas en blanco
  - Usar espacios en blanco para mejorar la legibilidad del código.
  - Usar espacios en blanco después de comas y después de las declaraciones.
  - Usar líneas en blanco para separar trozos de código de alta complejidad.
- Sustitución por Variables
  - Cuando un texto contiene porciones que serán sustituidas por el valor de alguna variable, se toma como correcto únicamente esta forma:  
  
`$saludo = "Hola $nombre, bienvenido!"`
- Concatenación
  - Para concatenar Strings se utilizará el operador "." (punto), con un espacio entre medio para mejorar la lectura.
- Arrays
  - En aquellos arrays de índices numéricos estos deberán ser números positivos, en la declaración de los valores del array se dejará un espacio en blanco luego de la coma para mejorar la lectura.
- Control de flujo
  - En las declaraciones if/else deberá tener un espacio antes y después del paréntesis condicional, en la misma línea se abre llave y se cierra en una línea diferente, lo mismo se aplica al elseif.

# Ampliación del Sistema

---

- En el caso de un Switch la convención es la misma, adicionando que el break tiene una indentación mayor que el resto.

Mediante el empleo de un estilo de programación para la confección de los subsistemas de Programación y Soporte es posible obtener un código más legible, permitiendo de esta forma la reducción de errores lógicos en el código y su reutilización.

## 2.7 Conclusiones Parciales

En este capítulo se expusieron aspectos esenciales que validan la ampliación del sistema, siendo el modelo de implementación y los diagramas de componentes los elementos a destacar, que posibilitan una mejor construcción y comprensión de cada uno de los componentes que integran el sistema ampliado. Además se describió y se caracterizó cada una de las funcionalidades de los subsistemas implicados, haciendo énfasis en cada uno de sus módulos, con el fin de esclarecer su funcionamiento. También se definieron y se caracterizaron los estándares de codificación que se utilizaron en la implementación de los módulos a ampliar, con el objetivo de asegurar que el código sea de alta calidad y que contenga el menor número de errores, para servir de ayuda y guía a otros desarrolladores que trabajen en nuevas versiones del sistema.

# Capítulo 3

## Validación de la solución propuesta

### 3.1 Introducción

En el presente capítulo se realiza un estudio de los diferentes métodos y tipos de pruebas con el objetivo de llevar a cabo la ejecución de las mismas, garantizando de esta forma la calidad del sistema y el total cumplimiento de los requisitos establecidos con el cliente.

El proceso de prueba es clave a la hora de detectar errores o fallas. Conceptos como estabilidad, escalabilidad, eficiencia y seguridad se relacionan con la calidad de un producto bien desarrollado. Las aplicaciones de software han crecido en complejidad y tamaño y por consiguiente en costos, por lo que se hace necesario que el proceso de pruebas se lleve a cabo con calidad y eficiencia, logrando la aceptación del producto por parte del usuario final.

### 3.2 Métodos de prueba

La prueba es un elemento crítico para la garantía de la calidad del software, es el proceso que permite verificar y revelar la calidad de un producto de software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un programa. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. La etapa de pruebas implica:

- ✓ Verificar la interacción de componentes.

# Validación de la solución propuesta

---

- ✓ Comprobar la integración adecuada de los componentes.
- ✓ Verificar que todos los requisitos se han implementado correctamente.
- ✓ Identificar y asegurar que los defectos encontrados se han corregido antes de entregar el software al cliente.
- ✓ Diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo.

Las pruebas de software consisten en una serie de actividades en las que un sistema o componente es ejecutado bajo determinadas condiciones o requerimientos, los resultados son observados y registrados, realizando una evaluación del sistema o del componente probado. Su principal objetivo es evaluar o valorar la calidad del producto.

Las pruebas se enfocan sobre la lógica interna del software y las funciones externas, para lo cual se definen los siguientes métodos:

## **3.2.1 Prueba de Caja Blanca**

Las pruebas de caja blanca realizan un seguimiento del código fuente según se van ejecutando los casos de prueba, de manera que se determinan de manera concreta las instrucciones en las que existen errores. Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas del código.

## **3.2.2 Prueba de Caja Negra**

Pruebas que se llevan a cabo sobre la interfaz de usuario, se centra principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Examina aspectos del modelo, principalmente del sistema, sin tener en cuenta la estructura interna del software.

## 3.3 Niveles de pruebas

A la hora de evaluar dinámicamente un sistema de software, la estrategia seleccionada debe permitir comenzar por los componentes más simples y pequeños e ir avanzando progresivamente hasta probar todo el sistema en su conjunto.

Para lograr una mayor efectividad, las pruebas de software se realizan a diferentes niveles:

**Prueba de desarrollador:** *La prueba de desarrollador indica los aspectos de diseño e implementación de las pruebas más adecuadas que debe llevar a cabo el equipo de desarrolladores, a diferencia de la prueba independiente. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas de desarrollador que la diseñó e implementó; aunque es recomendable que los desarrolladores creen las pruebas de forma que estén disponibles para que las ejecuten grupos de pruebas independientes.*

**Prueba independiente:** *Las pruebas independientes indican el diseño y la implementación de la prueba realizada más adecuadamente por alguien ajeno al equipo de desarrolladores. Puede considerar esta distinción un súper conjunto, que incluye validación y verificación independientes. En la mayoría de los casos, la ejecución de la prueba se produce inicialmente con el grupo de pruebas independientes que la diseñó e implementó; aunque los verificadores independientes deberían crear sus pruebas de forma que estén disponibles para que las ejecuten los grupos de pruebas de desarrollador.*

**Prueba de unidad:** *La prueba de unidad se centra en la verificación de los elementos más pequeños del software que se puedan probar. Normalmente, las pruebas de unidad se aplican a componentes representados en el modelo de implementación para verificar que se cubren los flujos de control y los flujos de datos y que funcionan como se esperaba. El implementador realiza la prueba de unidad mientras se desarrolla la unidad. Los detalles de la prueba de unidad se describen en la disciplina de implementación.*

**Prueba de integración:** *Las pruebas de integración se realizan para garantizar que los componentes del modelo de implementación funcionan correctamente cuando se combinan para ejecutar un guión de uso. El destino de la prueba es un paquete o un conjunto de paquetes del*

# Validación de la solución propuesta

---

*modelo de implementación. A menudo, los paquetes que se combinan proceden de diferentes empresas de desarrollo. Las pruebas de integración exponen el estado incompleto o los errores de las especificaciones de la interfaz del paquete.*

*En algunos casos, los desarrolladores presuponen que otros grupos, como los verificadores independientes, son los encargados de realizar las pruebas de integración. Esta situación supone un riesgo para el proyecto de software y, en última instancia, para la calidad del software, puesto que:*

- *Las áreas de integración son un punto común de fallo del software.*
- *Las pruebas de integración realizadas por verificadores independientes suelen utilizar técnicas de caja negra y tratar componentes de software más grandes.*

**Prueba del sistema:** *Normalmente, la prueba del sistema se realiza cuando el software funciona en su totalidad. Un ciclo vital repetitivo permite que las pruebas del sistema se realicen mucho antes, en cuanto se hayan implementado subconjuntos bien formados del comportamiento de guiones de uso. Normalmente, el destino son los elementos en funcionamiento de extremo a extremo del sistema.*

**Prueba de aceptación:** *La prueba de aceptación del usuario es la última acción de prueba antes de desplegar el software. El objetivo de la prueba de aceptación es comprobar si el software está preparado y lo pueden utilizar los usuarios para realizar las funciones y tareas para las que se diseñó. (1)*

## 3.4 Tipos de Pruebas

Con el objetivo de lograr que las pruebas aplicadas a un software impliquen mucho más que la simple evaluación de las funciones, la interfaz y las características de tiempo de respuesta, se aplican además un conjunto de pruebas adicionales que certifican su integridad, capacidad para instalarse y ejecutarse en plataformas diferentes capacidad para manejar varias solicitudes al mismo tiempo.

A continuación se ilustran los tipos de pruebas basados en las dimensiones de calidad.

# Validación de la solución propuesta

Tabla 1 Tipos de Pruebas que se aplican al software

Tipos de Pruebas que se aplican al software (1)	
Dimensión de calidad	Tipo de pruebas
Funcionalidad	<b>Prueba de función:</b> pruebas que se centran en la validación de funciones del destino de la prueba, proporcionan los guiones de uso, los métodos y los servicios necesarios. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades, las unidades integradas, las aplicaciones y los sistemas.
	<b>Prueba de seguridad:</b> pruebas que se centran en garantizar que los datos del destino de la prueba (o sistemas) sólo son accesibles para los actores a los que se dirigen. Esta prueba se implementa y ejecuta en varios destinos de la prueba.
	<b>Prueba de volumen:</b> pruebas que se centran en la verificación de la capacidad del destino de la prueba para manejar grandes cantidades de datos, ya sean de entrada y salida o residentes, en la base de datos. La prueba de volumen incluye estrategias de prueba como la creación de consultas que devolverán el contenido completo de la base de datos, o que tendrán tantas restricciones que no devolverán ningún dato, o en las que la entrada de datos tiene la cantidad máxima de datos para cada campo.
	<b>Prueba de integridad:</b> pruebas que se centran en la evaluación de la fuerza del destino de la prueba (resistencia a los errores) y la conformidad técnica del lenguaje, la sintaxis y la utilización de recursos. Esta prueba se implementa y se ejecuta en diferentes destinos de la prueba, incluidas las unidades y las unidades integradas.

# Validación de la solución propuesta

---

Fiabilidad	<p><b>Prueba de estructura:</b> pruebas que se centran en la evaluación de la adherencia del destino de la prueba a su diseño y formación. Normalmente, esta prueba se realiza en aplicaciones habilitadas para web y garantiza que todos los enlaces están conectados, se muestra el contenido adecuado y no hay ningún contenido huérfano.</p>
	<p><b>Prueba de tensión:</b> se trata de un tipo de prueba de fiabilidad que se centra en la evaluación de cómo responde el sistema en circunstancias anormales. Las tensiones del sistema pueden ser cargas de trabajo extremas, memoria insuficiente, servicios y hardware no disponible o recursos compartidos limitados. Estas pruebas suelen realizarse para saber mejor cómo y en qué áreas fallará el sistema, de forma que se puedan planificar y presupuestar los planes de contingencia y el mantenimiento de las actualizaciones con bastante antelación.</p>
	<p><b>Prueba de puntos de referencia:</b> se trata de un tipo de prueba de rendimiento que compara el rendimiento de un destino de la prueba nuevo o desconocido con una referencia conocida, carga de trabajo y sistema.</p>
	<p><b>Prueba de contienda:</b> pruebas que se centran en la validación de la capacidad del destino de la prueba para manejar de forma aceptable varias demandas del actor en el mismo recurso (registros de datos, memoria, etc.).</p>
	<p><b>Prueba de carga:</b> se trata de un tipo de prueba de rendimiento que se utiliza para validar y evaluar la aceptabilidad de los límites operativos de un sistema bajo cargas de trabajo variables, mientras el sistema que se está probando permanece igual. En algunas variantes, la carga de trabajo permanece igual y se modifica la configuración del sistema que se está probando. Las medidas suelen tomarse en función del</p>

# Validación de la solución propuesta

---

Rendimiento	<p>rendimiento de la carga de trabajo y el tiempo de respuesta de las transacciones en línea. Las variaciones de la carga de trabajo suelen incluir la emulación del pico y el promedio de cargas de trabajo que se producen dentro de la tolerancia operativa normal.</p>
	<p><b>Perfil de rendimiento:</b> se trata de una prueba en la que se controla el perfil de tiempo del destino de la prueba, incluidos el flujo de la ejecución, el acceso de datos, las llamadas del sistema y de funciones para identificar y tratar los cuellos de botella de rendimiento y los procesos ineficaces.</p>
	<p><b>Prueba de configuración:</b> pruebas que se centran en garantizar que las funciones del destino de la prueba son las adecuadas en diferentes configuraciones de hardware y software. Esta prueba también se puede implementar como una prueba de rendimiento del sistema.</p>
Capacidad de soporte	<p><b>Prueba de instalación:</b> pruebas que se centran en garantizar que el destino de la prueba se instala correctamente en diferentes configuraciones de hardware y software, y en condiciones diferentes (como, por ejemplo, espacio de disco insuficiente o interrupciones de la alimentación). Esta prueba se implementa y ejecuta en aplicaciones y sistemas.</p>

## 3.5 Diseño de las pruebas

Con el objetivo de validar los módulos desarrollados pertenecientes al Subsistema de Equipamiento y de Programación respectivamente, se realizaron un conjunto de pruebas funcionales que tienen

# Validación de la solución propuesta

---

por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados.

*A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que los probadores o analistas de pruebas, no enfocan su atención a como se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas. (2)*

A continuación se muestran los casos de pruebas correspondientes a los subsistemas desarrollados:

## 3.5.1 Casos de pruebas para el Módulo Gestionar Equipamiento.

**Tabla 2** CU\_ Adicionar Equipamiento

Nombre del Caso de Uso	Adicionar Equipamiento
Precondición	El usuario se debe haber autenticado como administrador.
Datos de entrada	Datos del nuevo equipamiento.
Resultados esperados	El sistema verifica los datos y los almacena en la base de datos. Muestra un mensaje de confirmación de que la operación se ah realizado correctamente.
	Si datos introducidos son incorrectos o se dejó en blanco algún campo obligatorio el sistema muestra un mensaje de error.
Resultados obtenidos	Satisfactorio
	Satisfactorio

# Validación de la solución propuesta

---

**Tabla 3 CU\_ Modificar Equipamiento**

Nombre del Caso de Uso	Modificar Equipamiento
Precondición	El usuario se debe haber autenticado como administrador.
Datos de entrada	El usuario selecciona el equipamiento a modificar.
Resultados esperados	El sistema actualiza, verifica los datos los datos y los almacena en la base de datos. Muestra un mensaje de confirmación de que la operación se ah realizado correctamente.
	Si datos introducidos son incorrectos o se dejó en blanco algún campo obligatorio el sistema muestra un mensaje de error.
Resultados obtenidos	Satisfactorio
	Satisfactorio

**Tabla 4 CU\_ Eliminar Equipamiento**

Nombre del Caso de Uso	Eliminar Equipamiento
Precondición	El usuario se debe haber autenticado como administrador.
Datos de entrada	El usuario selecciona el equipamiento a eliminar.
Resultados esperados	El sistema elimina el equipamiento de la base de datos. Muestra un mensaje de confirmación de que la operación se ah realizado correctamente.
Resultados obtenidos	Satisfactorio

# Validación de la solución propuesta

---

	Satisfactorio
--	---------------

**Tabla 5 CU\_ Buscar Equipamiento**

Nombre del Caso de Uso	Buscar Equipamiento
Precondición	El usuario se debe haber autenticado como administrador.
Datos de entrada	El usuario inserta el criterio de búsqueda.
Resultados esperados	El sistema visualiza el equipo existente.
	El sistema muestra un mensaje de error que indica que no se encontró ningún equipo.
Resultados obtenidos	Satisfactorio
	Satisfactorio

## 3.5.2 Casos de pruebas para el Módulo Gestión de la Programación.

**Tabla 6 CU\_ Adicionar Plan de Programación**

Nombre del Caso de Uso	Adicionar Plan de Programación
Precondición	El usuario se debe haber autenticado como administrador.
Datos de entrada	Datos de la programación correspondientes a un plan de programación para una fecha y un canal determinados.

## Validación de la solución propuesta

Resultados esperados	El sistema verifica los datos y los almacena en la base de datos. Muestra un mensaje de confirmación de que la operación se ah realizado correctamente.
	Si datos introducidos son incorrectos o se dejó en blanco algún campo obligatorio el sistema muestra un mensaje de error.
Resultados obtenidos	Satisfactorio
	Satisfactorio

**Tabla 7 CU\_ Modificar Plan de Programación**

Nombre del Caso de Uso	Modificar Plan de Programación
Precondición	El usuario se debe haber autenticado como administrador.
Datos de entrada	El usuario selecciona el plan de programación a modificar.
Resultados esperados	El sistema actualiza, verifica los datos y los almacena en la base de datos. Muestra un mensaje de confirmación de que la operación se ah realizado correctamente.
	Si datos introducidos son incorrectos o se dejó en blanco algún campo obligatorio el sistema muestra un mensaje de error.
Resultados obtenidos	Satisfactorio
	Satisfactorio

# Validación de la solución propuesta

**Tabla 8 CU\_ Eliminar Plan de Programación**

Nombre del Caso de Uso	Eliminar Plan de Programación
Precondición	El usuario se debe haber autenticado como administrador.
Datos de entrada	El usuario selecciona, la programación a eliminar
Resultados esperados	El sistema elimina la programación de la base de datos. Muestra un mensaje de confirmación de que la operación se ah realizado correctamente.
	Si se eliminan todas las programaciones correspondientes a un plan de programación para un canal especifico, se elimina ademas dicho plan. Muestra un mensaje de confirmación de que la operación se ah realizado correctamente.
Resultados obtenidos	Satisfactorio
	Satisfactorio

**Tabla 9 CU\_ Visualizar Plan de Programación**

Nombre del Caso de Uso	Visualizar Plan de Programación
Precondición	El usuario se debe haber autenticado como administrador.
Datos de entrada	Se selecciona una fecha y un canal respectivamente.
Resultados esperados	El sistema muestra el plan de programación correspondiente.
Resultados obtenidos	Satisfactorio

## Validación de la solución propuesta

---

La realización de las pruebas de funcionalidad a los subsistemas incluidos al SGPDTU arrojaron un resultado de satisfactorio en cada uno de los casos de pruebas definidos. También se realizaron pruebas de stress, las cuales arrojaron como resultado que el sistema se demora más de lo establecido para cargar con un número superior a los 40 usuarios conectados simultáneamente, por lo que se debe tener en cuenta que el servidor donde se va a montar la aplicación debe tener buenas condiciones técnicas. Además se le aplicaron de manera análoga las pruebas de Unidad e Integración, además se le designó una prueba de Aceptación donde la DTU quedó satisfecha con las nuevas modificaciones en el sistema.

## **3.6 Conclusiones Parciales**

Con la puesta en práctica de una serie de pruebas se validó que cada uno de los módulos implementado correspondía con las descripciones de los casos de usos y los requisitos funcionales establecidos, quedando como constancia el correcto funcionamiento de cada una de las funcionalidades, así como su calidad.

## CONCLUSIONES GENERALES

El presente Trabajo de Diploma posibilitó la implementación de nuevos módulos en el Sistema de Gestión para la Dirección de Televisión Universitaria (SGPDTU), considerándose como una ampliación del mismo y se concluye lo siguiente:

- Se solventan mayoritariamente todas las necesidades que presentaba el sistema inicial, quedando bien documentadas cada una de las actividades realizadas para su implementación.
- El módulo de Gestión de la Programación crea una cartelera televisiva de forma dinámica para cada uno de los canales existentes en la UCI, con las medias definidas, además crea un documento Adobe Reader con la programación completa para cada canal en una fecha especificada.
- El módulo de Gestión del Equipamiento lleva el control de todo el equipamiento técnico de la DTU como son cámaras, cableado, micrófonos, entre otros, además se integra al módulo de producción, el cual para su funcionamiento necesita conocer los equipamientos disponibles y poder asignarlos en dependencia del plan de producción.
- La realización de pruebas de calidad de software a la aplicación con el objetivo de validar la propuesta de la solución, arrojaron resultados favorables al sistema, pruebas de funcionalidad, estrés, unidad, integración y aceptación revelaron un sistema limpio y cómodo para los usuarios.

## RECOMENDACIONES

Luego de concluir la implementación del sistema y realizarle las suficientes pruebas al mismo, se recomienda:

- Continuar añadiendo funcionalidades y módulos al SGPDTU que contribuyan a su mejor uso.
- Hacer énfasis y profundizar en el estudio de las librerías Dojo utilizadas en el sistema, pues las mismas permiten una mejor visualización y una interfaz de usuario más amigable, además de un alto nivel en cuanto al manejo de los datos.
- Valorar las tecnologías aplicadas a la solución propuesta en pos de su utilización en otros proyectos de la Universidad.

# GLOSARIO DE TÉRMINOS

**Media:** Conjunto de los medios de comunicación; dígase sonido, gráficos, animación y vídeo, que son utilizados en la transmisión de información. Oriundo del idioma inglés (mass media).

**URL (Uniform Resource Locator):** es una dirección que permite acceder a un archivo o recurso como ser páginas html, php, asp, o archivos gif, jpg, etc. Se trata de una cadena de caracteres que identifica cada recurso disponible en la WWW.

**XML (Extensible Markup Language):** Es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados. En la práctica corresponde a un estándar que permite a diferentes aplicaciones interactuar con facilidad a través de La Red.

**XHTML:** es una familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0. Los tipos de documentos de la familia XHTML están basados en XML, y diseñados fundamentalmente para trabajar en conjunto con aplicaciones de usuario basados en XML.

**XSLT:** transforma documentos XML en documentos XHTML u otros documentos XML. El W3C es el encargado de la definición de especificación XSLT. XSLT se basa en XPath para realizar la búsqueda de información a través del documento XML. XPath son cadenas que son expresiones regulares, las cuales hacen referencia a alguna estructura dentro del documento XML. Introducción a XSLT o XSL Transformations, la parte más importante del lenguaje XSL (eXtensible StyleSheet Language).

**XMLHttpRequest:** es un API que puede invocarse desde JavaScript, y otros lenguajes de script incluidos en un navegador web, que se usa para transferir y manipular datos XML hacia y desde el navegador web, estableciéndose un canal de conexión independiente entre el lado del cliente de la página web y el servidor.

**CSS (Cascading Style Sheets):** es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de

# GLOSARIO DE TÉRMINOS

---

descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

**DOM (Document Object Model):** es la estructura de objetos que genera el navegador cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página.

**SGML:** es el acrónimo de Standard Generalized Markup Language es un estándar internacional, o sea, el ISO 8879:1986 utilizado para la definición formal de documentos con modalidad que lo hace ser independiente del dispositivo, sistema y aplicación con el cual estos documentos vienen realizados; en otras palabras, SGML es un metalenguaje, o sea un lenguaje usado para describir otros lenguajes.

**PLUG-IN:** es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

# REFERENCIAS BIBLIOGRÁFICAS

1. **IBM Corporation.** Ayuda Rational Unified Process,2007. *Tipos de prueba.* [En línea] 2006. [Citado el: 3 de Mayo de 2010.]

Ayuda.del.RUP.2007.Español/core.base\_rup/guidances/concepts/types\_of\_test\_8AB94831.html.

2. **Oré B., Ing. Alexander.** CalidadSoftware.com. *FUNCTIONAL TESTING - PRUEBAS FUNCIONALES.* [En línea] 2009. [Citado el: 4 de Mayo de 2010.] [http://www.calidadsoftware.com/testing/pruebas\\_funcionales.php](http://www.calidadsoftware.com/testing/pruebas_funcionales.php).

3. **Ramos Núñez, Yusbel.** *Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria.* 2009.

4. **NEWMAN, William H.** *Programación, Organización y Control.*

5. **Kirk Kanadt, Ronald.** *Software Engineering Quality Practices.* 2006.

6. **Junta de Castilla y León.** Glosario. [En línea] 2009.

<http://www.eclap.es/contenidos/calidad/GLOSARIO/GLOSARIO.doc>.

7. **Mendoza S., María A.** Informatizate. [En línea] 2004.

[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).

8. **Varios.** *Datapro Client/Server Analyst, cliente server computing: emerging trends, solutions and strategies.* 1994.

9. Arquitectura Cliente/Servidor. *CSAE.* [En línea] [www.csae.map.es](http://www.csae.map.es).

10. **VALENCIA.** *Proceso de desarrollo de software.*

11. **Jiménez Garzón, Darwin.** Ingeniería de Software II.

12. **Chuck Musciano, Bill Kenedy.** HTML. La Guía Completa. [En línea] 2009.

<http://bibliodoc.uci.cu/pdf/reg01313.pdf>.

13. **Frentzen, Jeff y Sobotka, Henry.** *Superutilidades para JavaScript.* 2003.

14. **Russell, Matthew A.** *Dojo The Definitive Guide.*

15. **Eguíluz Pérez, Javier.** Introducción a AJAX. *libros Web.* [En línea] 2008. [www.librosweb.es](http://www.librosweb.es).

# REFERENCIAS BIBLIOGRÁFICAS

---

16. **Rational White Pape.** *Best Practices for Software Development Teams.* 1998.
17. **Hernández Díaz, Neysis y Castro Sierra, Frank Alain.** PLICACIÓN WEB PARA GESTIONAR Y MONITOREAR LA MIGRACIÓN DE DATOS DEL SISTEMA ASSETS AL SISTEMA TRABAJADORES EN LA UCI. UCI : s.n., 2007.
18. **Achour, Mehdi, Betz, Friedhelm y Dovgal, Antony.** Grupo de documentación de PHP. *PHP.* [En línea] 2009. <http://www.php.net/manual/es/index.php>.
19. **Garcia de Jalon, Javier.** Aprenda Java como si estuviera en primero. San Sebastian : Escuela Superior de Ingenieros Industriales : s.n., 2000.
20. **Potencier, Fabien y Zaninotto, François.** Symfony la guía definitiva. *libros Web.* [En línea] 2008. [www.librosweb.es](http://www.librosweb.es).
21. **GUTIÉRREZ, G. A.** J2EE Una plataforma para el cómputo empresarial. 2003.
22. **Eclipse.** Eclipse. *Eclipse.* [En línea] 2009. <http://www.eclipse.org/>.
23. **NetBeans.** *De Guía Ubuntu.* [En línea] <http://netbeans.org/kb/>.

# BIBLIOGRAFÍA

1. **IBM Corporation.** Ayuda Rational Unified Process,2007. *Tipos de prueba.* [En línea] 2006. [Citado el: 3 de Mayo de 2010.]

Ayuda.del.RUP.2007.Español/core.base\_rup/guidances/concepts/types\_of\_test\_8AB94831.html.

2. **Oré B., Ing. Alexander.** CalidadSoftware.com. *FUNCTIONAL TESTING - PRUEBAS FUNCIONALES.* [En línea] 2009. [Citado el: 4 de Mayo de 2010.] [http://www.calidadsoftware.com/testing/pruebas\\_funcionales.php](http://www.calidadsoftware.com/testing/pruebas_funcionales.php).

3. **Ramos Núñez, Yusbel.** *Sistema de Gestión de Procesos para la Dirección de Televisión Universitaria.* 2009.

4. **NEWMAN, William H.** *Programación, Organización y Control.*

5. **Kirk Kanadt, Ronald.** *Software Engineering Quality Practices.* 2006.

6. **Junta de Castilla y León.** Glosario. [En línea] 2009.

<http://www.eclap.es/contenidos/calidad/GLOSARIO/GLOSARIO.doc>.

7. **Mendoza S., María A.** Informatizate. [En línea] 2004.

[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html).

8. **Varios.** *Datapro Client/Server Analyst, cliente server computing: emerging trends, solutions and strategies.* 1994.

9. Arquitectura Cliente/Servidor. *CSAE.* [En línea] [www.csae.map.es](http://www.csae.map.es).

10. **VALENCIA.** *Proceso de desarrollo de software.*

11. **Jiménez Garzón, Darwin.** *Ingeniería de Software II.*

12. **Chuck Musciano, Bill Kenedy.** HTML. La Guía Completa. [En línea] 2009.

<http://bibliodoc.uci.cu/pdf/reg01313.pdf>.

13. **Frentzen, Jeff y Sobotka, Henry.** *Superutilidades para JavaScript.* 2003.

14. **Russell, Matthew A.** *Dojo The Definitive Guide.*

15. **Eguíluz Pérez, Javier.** Introducción a AJAX. *libros Web.* [En línea] 2008. [www.librosweb.es](http://www.librosweb.es).

# BIBLIOGRAFÍA

---

16. **Rational White Pape.** *Best Practices for Software Development Teams.* 1998.
17. **Hernández Díaz, Neysis y Castro Sierra, Frank Alain.** PLICACIÓN WEB PARA GESTIONAR Y MONITOREAR LA MIGRACIÓN DE DATOS DEL SISTEMA ASSETS AL SISTEMA TRABAJADORES EN LA UCI. UCI : s.n., 2007.
18. **Achour, Mehdi, Betz, Friedhelm y Dovgal, Antony.** Grupo de documentación de PHP. *PHP.* [En línea] 2009. <http://www.php.net/manual/es/index.php>.
19. **García de Jalon, Javier.** Aprenda Java como si estuviera en primero. San Sebastian : Escuela Superior de Ingenieros Industriales : s.n., 2000.
20. **Potencier, Fabien y Zaninotto, François.** Symfony la guía definitiva. *libros Web.* [En línea] 2008. [www.librosweb.es](http://www.librosweb.es).
21. **GUTIÉRREZ, G. A.** J2EE Una plataforma para el cómputo empresarial. 2003.
22. **Eclipse.** Eclipse. *Eclipse.* [En línea] 2009. <http://www.eclipse.org/>.
23. NetBeans. *De Guía Ubuntu.* [En línea] <http://netbeans.org/kb/>.
24. **Eguíluz Pérez, Javier.** Introducción a CSS. *libros Web.* [En línea] 2009. [www.librosweb.es](http://www.librosweb.es).
25. **Pérez Mira, Reynier.** SYMFONY: Framework para desarrollo de aplicaciones con PHP. 2009.
26. **Pérez, Javier Eguíluz.** Introducción a XHTML. *libros Web.* [En línea] 2009. [www.librosweb.es](http://www.librosweb.es).
27. **Ribamar FS.** PostgreSQL Práctico. [En línea] 2006. <http://ribafs.tk> 2006.
28. PHP. *ciberaula.* [En línea] [http://php.ciberaula.com/articulo/introduccion\\_php](http://php.ciberaula.com/articulo/introduccion_php).
29. PHP. *desarrollo web.* [En línea] <http://www.desarrolloweb.com/articulos/392.php>.
30. Eclipse. *desarrollo web.* [En línea] <http://www.desarrolloweb.com/articulos/pdt-eclipse-php.html>.
31. Framework. *css blog.* [En línea] <http://www.cssblog.es/guias/Framework.pdf>.
32. Ubuntu. [En línea] <http://www.compralomx.com/blog.php?post=75>.
33. Apache. *abc datos.* [En línea] <http://www.abcdatos.com/webmasters/programa/z2820.html>.

Ciudad de la Habana, Junio 2010

“Año 52 de la Revolución”