

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

ANÁLISIS, DISEÑO Y CONSTRUCCIÓN DEL PORTAL WEB CON ACCESO A BASE DE DATOS PARA LA SOCIEDAD DE AUTORES Y COMPOSITORES ECUATORIANOS

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO, EN
ANÁLISIS DE SISTEMAS INFORMÁTICOS**

KATHERINE DEL ROCÍO GOYES MOLINA

katherineg@sayce.com.ec

HÉCTOR ARTURO MONTENEGRO MONTENEGRO

hectormontenegro@unita.edu.ec

DIRECTOR: Ing. MIRIAN PEÑAFIEL AGUILAR MSc.

mpenafiel@gmail.com

Quito, Octubre del 2008

DECLARACIÓN

Nosotros, Katherine del Rocío Goyes Molina y Héctor Arturo Montenegro Montenegro, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos los derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Katherine del Rocío Goyes Molina Héctor Arturo Montenegro Montenegro

CERTIFICACIÓN

Certifico que el presente trabajo de tesis, fue desarrollado en su totalidad por Katherine del Rocío Goyes Molina y Héctor Arturo Montenegro Montenegro, bajo mi supervisión.

Ing. Miriam Peñafiel
DIRECTOR DE PROYECTO

AGRADECIMIENTO

Queremos aprovechar este espacio para agradecer primero a Dios por habernos dado la oportunidad que muy pocos la tienen y que es la de estudiar.

Gracias a nuestras madres y hermanos que con su invaluable amor y apoyo siempre supieron guiarnos para salir adelante, gracias por la paciencia y la constancia que día a día nos inculcaron.

Gracias profesores por darnos sus enseñanzas y mostrarnos sus puntos de vista, por alentarnos, por ayudarnos, por esa amistad que muchos de ellos supieron darnos, por esa exigencia y costumbre de trabajar que nos inculcaron. Gracias por ser esos alentadores de espíritu, esos señores y señoras cultas que comparten su conocimiento y sus experiencias. Gracias amigos que pusieron ejemplo y camino a seguir y a veces a evitar, gracias por todo corazón.

DEDICATORIA

Queremos dedicar este trabajo a Dios que nos ha dado la vida y fortaleza para terminar este proyecto. A nuestros padres por estar ahí cuando más lo necesitamos; en especial a nuestras madres por su amor, ayuda y constante cooperación y a nuestros amigos por apoyarnos y ayudarnos en todo momento.

RESÚMEN

El presente proyecto tiene por objetivo el análisis, diseño y construcción del Portal Web para la SAYCE (Sociedad de Autores y Compositores Ecuatorianos), cuya finalidad será ofrecer servicios remotos vía Internet para proporcionar información a Socios y Usuarios en general vinculados a esta sociedad de gestión colectiva y además llevar el registro de Socios y Obras que administra SAYCE a nivel nacional, además permitirá realzar y poner a conocimiento de la Comunidad de la gestión que realiza SAYCE a nivel nacional e internacional, publicar noticias inherentes a la sociedad, publicar en forma dinámica a la comunidad sobre los Catálogos de Autores y/o Compositores, así como sus obras, biografías, publicar los datos relevantes de sus socios y sus obras, además la publicación de las tarifas por los diferentes conceptos de uso de música.

El Portal Web para la Sociedad de Autores y Compositores Ecuatorianos, tiene el objetivo de mantener informados a socios y usuarios sobre toda la actividad que realiza la SAYCE, como es la publicación de socios afiliados obras de su administración, y tarifas.

Este portal tendrá dos tipos de usuario, el primero que es un usuario que solo podrá acceder a la lectura de la información, el segundo podrá modificar los datos del catálogo con el fin de mantener actualizada la información de este portal.

Evidentemente, existirán nuevos requerimientos que necesitará la SAYCE, de ser posible se los desarrollarán a medida que avance el proyecto dejando estos procesos culminados o por lo menos planteados para un nuevo proyecto a manera del paradigma incremental.

CONTENIDO

CAPITULO 1

ANTECEDENTES	1
1.1 EL PROBLEMA	1
1.2 OBJETIVOS	1
1.2.1 OBJETIVO GENERAL	1
1.2.2 OBJETIVOS ESPECIFICOS	2
1.2.3 ALCANCE DEL PROYECTO	2
1.3 BREVE EXPLICACION DEL PROYECTO	3

CAPITULO 2

MARCO TEORICO	4
2.1 PARADIGMA Y METODOLOGIA: ORIENTADA A OBJETOS Y ADPD-OMG	4
2.2 EL CICLO DE VIDA ADPD	6
2.2.1 Fase Walking Skeleton (esqueleto guía)	10
2.2.2 Fase de liberación	12
2.3 PRÁCTICAS ADPD	12
2.3.1 Prácticas de Análisis y Diseño	13
2.3.2 Prácticas de desarrollo	17
2.3.3 Prácticas de test	19
2.3.4 Prácticas de implantación y liberación	23
2.3.5 PRACTICAS GENERALES Y DE DIRECCION	24
2.4 Adopción de ADPD y UML	30
2.5 UML en breve	31
2.6 UML y el Ciclo de vida ADPD	34
2.6.2 Fase de liberación	42

CAPITULO 3

HERRAMIENTAS	45
3.1 INTRODUCCION A LINUX	45
3.1.1 UN POCO DE HISTORIA	45
3.1.2 ¿QUE ES LINUX?	47
3.1.3 REQUISITOS DE LA PLACA BASE Y DEL PROCESADOR	49
3.1.4 REQUISITOS DE LA MEMORIA RAM	49
3.1.5 REQUISITOS DE ESPACIO EN EL DISCO DURO	50
3.1.6 CONTROLADOR DE DISPOSITIVOS	51
3.1.7 BUSES Y TIPOS DE PUERTOS SOPORTADOS POR LINUX	51
3.1.8 REQUISITOS DEL SISTEMA GRAFICO	51
3.1.9 SOPORTE DE DISPOSITIVOS DE CONEXIÓN A LA RED LOCAL	51
3.1.10 SOPORTE DE IMPRESORAS	51
3.1.11 ESTRUCTURA	51
3.2. ACERCA DEL PROTOCOLO HTTP.	52

3.3 APACHE.	52
3.3.1 ¿Qué es Apache?	52
3.3.2 ¿Cómo se comporta Apache en relación a las pruebas comparativas con otros servidores de la web?	53
3.3.3 ¿Por qué precisamente Apache con todos los servidores de Web que se encuentran en circulación?	53
3.3.4 ¿Qué hacer para configurar Apache?	54
3.4 MySQL	54
3.4.1 ¿Qué es MySQL?	54
3.4.2 Historia de MySQL	54
3.4.3 Características de MySQL	55
3.4.4 Lo mejor de MySQL	55
3.5 PHP	56
3.5.1 Historia de PHP	56
3.5.2 ¿Qué hace PHP?	57
3.6 PHPMYADMIN	58
3.7 FLASH	59
3.7.1 Línea de tiempo por fotogramas	60
3.7.2 Dibujos y formas	60
3.7.3 Desarrollo de ActionScript 2.0	60
3.7.4 Sofisticadas herramientas de vídeo	60
3.7.5 Amplias funciones de diseño	60
3.7.6 Arquitectura ampliable	61
3.8 DREAMWEAVER	61
3.8.1 Qué es Dreamweaver 8?	61
3.8.2 Crea contenido web sofisticado	61
3.8.3 Prácticas recomendadas	61
3.9 RATIONAL SYSTEMS DEVELOPER	62
3.10 CentOS 5.	62
3.10.1 CONSIDERACIONES PREVIAS A LA INSTALACION DE CENTOS 5	62
3.10.2 DISPOSITIVOS Y PARTICIONES EN LINUX	62
3.10.3 CARACTERISTICAS DEL HARDWARE USADO PARA LA INSTALACION:	65
3.10.4 INSTALACION	65
3.10.5 CONFIGURACION CENTOS	85
3.10.6 CONFIGURACION MySQL	87
3.10.7 CONFIGURACION APACHE Y PHP	88
 CAPITULO 4	
CONCLUSIONES Y RECOMENDACIONES	89
4.1 CONCLUSIONES	89
4.2 RECOMENDACIONES	89
4.3 Bibliografía:	90
 ANEXO A	92
 ANEXO B	142

CAPITULO 1

ANTECEDENTES

1.1 EL PROBLEMA

La Sociedad de Autores y Compositores Ecuatorianos SAYCE, es una Entidad de Gestión Colectiva de Derecho privado, con finalidad social; cuyo deber primordial es preservar los derechos morales y administrar los derechos patrimoniales (económicos) resultantes de la utilización pública de obras musicales de autores nacionales y/o extranjeros a los que representa.

Como Entidad Autoral es Miembro Ordinario de la Confederación Internacional de Sociedades de Autores y Compositores, C.I.S.A.C. Organismo Intercontinental para la protección de Derechos de Autor y de otros organismos afines.

La importancia de dar a conocer la gestión de la SAYCE, requiere de la construcción de un Sitio Web que permita a los socios nacionales, sociedades extranjeras y usuarios de la música conocer: bases legales, socios nacionales y extranjeros, tarifas en los diferentes conceptos de recaudación, repertorio administrado, y noticias inherentes a la gestión.

Este proyecto estará estructurado por fases iterativas que a su vez son incrementales. Cada una de las iteraciones tiene etapas de: análisis, diseño, desarrollo, test, implantación, liberación.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Diseñar e implementar el Portal SAYCE (Portal Web para la Sociedad de Autores y Compositores Ecuatorianos), cuya finalidad será ofrecer servicios remotos vía Internet para proporcionar información a Socios y Usuarios en general vinculados a esta sociedad de gestión colectiva y además llevar el registro de Socios y Obras que administra SAYCE a nivel nacional.

1.2.2 OBJETIVOS ESPECIFICOS

Realzar y poner en conocimiento a la comunidad de la gestión que realiza la SAYCE a nivel nacional e internacional.

Realizar publicaciones de eventos y noticias inherentes a la sociedad.

Publicación dinámica para informar a la comunidad sobre los Autores y/o compositores, así como sus obras.

Apoyar a la parte administrativa registrando los datos relevantes de sus socios y sus obras.

Publicación de tarifas por los diferentes conceptos de uso de música

1.2.3 ALCANCE DEL PROYECTO

Aclaremos que todos los procesos de este Portal tienen la característica de que la información pueden ser accedida desde cualquier lugar remoto, ya que estará publicado vía Internet.

El Portal SAYCE proporcionará la siguiente información:

Los usuarios podrán tener acceso a información de las diferentes actividades y servicios que presta SAYCE a la comunidad:

Organización

Bases Legales

Socios y obras

Filiales

Tarifas

Actualización de la información del Portal ya que se requiere mantener informada a la comunidad sobre las fechas importantes como son:

Condecoración a socios

Entrega de pensiones vitalicias

Realización de asambleas generales

Eventos internacionales

Al usuario se le permitirá realizar consultas remotas vía Internet de las obras y socios que hayan firmado el contrato de adhesión.

Ayudará a la parte administrativa a llevar un repositorio de datos y emitir reportes de socios afiliados a los diferentes consejos provinciales.

Publicación de tarifas por concepto de:

Espectáculos públicos.

Usuarios generales.

Reproducción fonográfica

Derechos de sincronización

Radio y televisión

El alcance del proyecto queda detallado, pero los temas que no consten en la descripción de este proyecto que lógicamente sean complementarios a la aplicación, serán implícitamente desarrollados.

1.3 BREVE EXPLICACION DEL PROYECTO

El Portal SAYCE para la Sociedad de Autores y Compositores Ecuatorianos, tiene el objetivo de mantener informados a socios y usuarios sobre toda la actividad que realiza SAYCE, publicación de obras de su administración, socios afiliados y tarifas.

Este portal tendrá dos tipos de usuario, el primero que es un usuario general que podrá acceder a la información, el segundo podrá modificar los datos del repositorio de datos con el fin de mantener actualizada la información de este portal.

Evidentemente, existirán nuevos requerimientos que necesitará la SAYCE, de ser posible se los desarrollarán a medida que avance el proyecto dejando estos procesos culminados o por lo menos planteados para un nuevo proyecto a manera del paradigma incremental.

CAPITULO 2

MARCO TEORICO

2.1 PARADIGMA Y METODOLOGÍA: ORIENTADO A OBJETOS Y ADPD-OMG

Ciclo De Vida: ADPD		Vistas OMG Metodología: ADPD-OMG	Diagramas UML2.0 Método: ADPD-UML2.0	Herramientas	Prácticas: ADPD
Fase Walking Skeleton (WS)	Análisis	Modelo del negocio	✓ Casos de Uso de Negocio ✓ Casos de Uso Textual	Rational Systems Developers	<ul style="list-style-type: none"> ○ Pequeñas liberaciones ○ Cliente en el sitio ○ Planificación del evento ○ Gerencia de requerimientos ○ Programación en par ○ Estándares de codificación ○ Sistema Versión Concurrente ○ Test Unitario ○ Test de Suite ○ Test Funcional ○ Testeo de Regresión
	Diseño	Modelo de estructura	✓ Clases ✓ Componentes ✓ Paquetes ✓ Implantación		
	Desarrollo	Modelo 3 capas (sólo estructura)	Presentación	HTML (embebido PHP)	
			Dominio (negocio)	PHP	
			Acceso a Datos	MySQL	
	Test	Qa Quality Assurance	Pruebas	Servidor LAMP (Linux, Apache, MySQL, PHP)	
Implantación/ Liberación			Servidor LAMP		
Fase de Liberación 1	Análisis	Modelo del negocio	✓ Casos de Uso de Negocio ✓ Casos de Uso Textual	Rational Systems Developers	<ul style="list-style-type: none"> ○ Pequeñas liberaciones ○ Cliente en el sitio ○ Planificación del evento ○ Gerencia de requerimientos ○ Programación en par ○ Estándares de codificación ○ Sistema de Versión Concurrente ○ Test Unitario ○ Test de Suite ○ Test Funcional ○ Testeo de Regresión
	Diseño	Modelo de estructura	✓ Clases ✓ Componentes ✓ Paquetes ✓ Implantación		
		Modelo de comportamiento	✓ Casos de Uso de Sistema ✓ Actividades		
	Desarrollo	Modelo 3 capas (Funcionalidad1)	Presentación	HTML (embebido PHP)	
			Dominio (negocio)	PHP	
			Acceso a Datos	MySQL	
Test	Qa Quality Assurance	Pruebas	Servidor LAMP		
Implantación/ Liberación			Servidor LAMP		
Fase de Liberación 2 * Funcionalidad 2					
:					
:					
Fase de Liberación n-1 * Funcionalidad n-1					
Fase de Liberación n * Proyecto concluido					

El paradigma Agile and Defined Project Development, del autor Fabricio Fioravanti, que en adelante lo llamaremos sólo por sus siglas ADPD será usado en el presente proyecto. El autor define las fases: Walking Skeleton, que en adelante la llamaremos sólo por sus siglas WS; y Liberación, ésta última es repetitiva (iterativa) hasta alcanzar la culminación de todas las funcionalidades (modularidades) del proyecto. Cada fase tiene sub-fases: análisis, diseño, desarrollo, test, implantación y liberación.

The Object Management Group (OMG) es un consorcio internacional sin fines de lucro. OMG impulsa la tarea del desarrollo de estándares de integración empresarial para el ancho grupo de tecnologías y un ancho grupo de industrias. Los estándares de modelamiento de OMG permite poderoso diseño visual, ejecución y mantenimiento del software y otros procesos.

The Unified Modeling Lenguaje (UML) es la especificación más usada de OMG, y la manera de modelar el mundo no sólo estructura, comportamiento y arquitectura de aplicaciones, sino también procesos de negocios y estructuras de datos.

Los modelos y diagramas sugeridos para las fases y sub-fases de ADPD, de cierta forma son comparados con los propuestos por OMG-UML. Estos ayudarán a especificar, visualizar y documentar los modelos del sistema de software. Los resultados de la solución se representan usando los 13 diagramas estándares de OMG-UML que a continuación son listados (los diagramas adoptados y sugeridos por ADPD están con un visto, pero con la aclaración de que si el sistema aún no es claro en su comprensión, ADPD propone que se deben añadir los diagramas necesarios para su total comprensión).

Diagramas de Estructura	<ul style="list-style-type: none"> ✓ Clases ✗ Objetos ✓ Componentes ✗ Estructura Compuesta ✓ Paquetes ✓ Implantación
Diagramas de Comportamiento	<ul style="list-style-type: none"> ✓ Casos de Uso(Negocio y Sistema) ✓ Actividades ✓ Maquinas de Estado
Diagramas de Interacción	<ul style="list-style-type: none"> ✗ Secuencia ✗ Comunicación < Colab ✗ Tiempos ✗ General de Interacción

La metodología Agile and Defined Project Development (ADPD), que significa Desarrollo de Proyecto Ágil y Definido, será presentada y discutida. El nombre y la consecuencia del acrónimo son debido al hecho que el autor quería crear una metodología que sea (A) ágil y, por lo tanto, obedece al Agile Manifesto, pero que al mismo tiempo puede ser ampliamente aceptada y también implantada en organizaciones que no se inclinan a aceptar el desarrollo Agile.

El principal propósito de ADPD es eliminar el criticismo que frecuentemente emergen de las metodologías Agile con fraccionamiento o desarrollo no estructurados. Para obtener tales resultados la metodología debe estar de acuerdo al menos con el Modelo de Capacidad y Madurez del Software (Software Capability Maturity Model), comúnmente conocido como CMM-SW (Paulk, 1993, 1933a) Level3: el nivel definido. Esto explica el segundo término del acrónimo (*D*). El tercero (*P*) y cuarto término (*D*) son bastante obvios y no necesitan mayor investigación.

En conformidad con CMM-SW Level3 permite exitosamente aplicar esta metodología en un ambiente donde una estandarización en términos de software o producto de calidad es indispensable, a partir del nivel definido de CMM-SW permite concordar los requerimientos de la mayoría de las compañías que usualmente no están de acuerdo con la metodología y gerencia Agile.

Por lo antes expuesto el autor empezó a definir una nueva metodología, principalmente basada en el concepto de la primera metodología Agile que se ha aplicado en proyectos reales, esta es, XP (Beck, 1999, 2000).

El autor ha insertado en ADPD todos los aspectos positivos y técnicas que son parte de XP y que fueron capaces de aplicar exitosamente en la diaria dirección de proyectos.

2.2 EL CICLO DE VIDA ADPD

La aplicación satisfactoria de una metodología es debida a la comprensión de su ciclo de vida y de las ventajas y los beneficios que la metodología ofrece o le propone conllevar.

ADPD ha sido creado con una principal finalidad en mente: Reducción de los costos de mantenimiento. En una publicación anterior por el mismo editor, se ha evidenciado cómo la metodología Agile y particularmente, la aplicación de XP pueden traer constantes costos de mantenimiento (Fioravanti, 2003) durante el tiempo de vida del proyecto, y se ha tratado de transferir estos aspectos hacia

ADPD con la finalidad de evitar una de las principales desventajas de las metodologías clásicas que consiste en el hecho de que el costo de mantenimiento no es una constante durante el ciclo de vida del proyecto. Esto es debido a la organización del ciclo de vida en bloques monolíticos grandes, donde, si un error ocurre en algún punto, esto exige replantear una parte grande del trabajo ya hecho.

Podemos evidenciar este hecho considerando diferentes ciclos de vida, tanto el ciclo de vida en cascada monolítico y el ciclo de vida espiral iterativo, comparado al Agile (con referencia particular para ADPD). Los pasos que son presentados en cada ciclo de vida pueden ser considerados los mismos o al menos muy similares. Lo que realmente difiere es el instante de la aplicación de cada paso y la consecuente organización de esos pasos. Para cada metodología, se asume tener los siguientes pasos: *análisis, diseño, desarrollo, test, e implantación/liberación.*

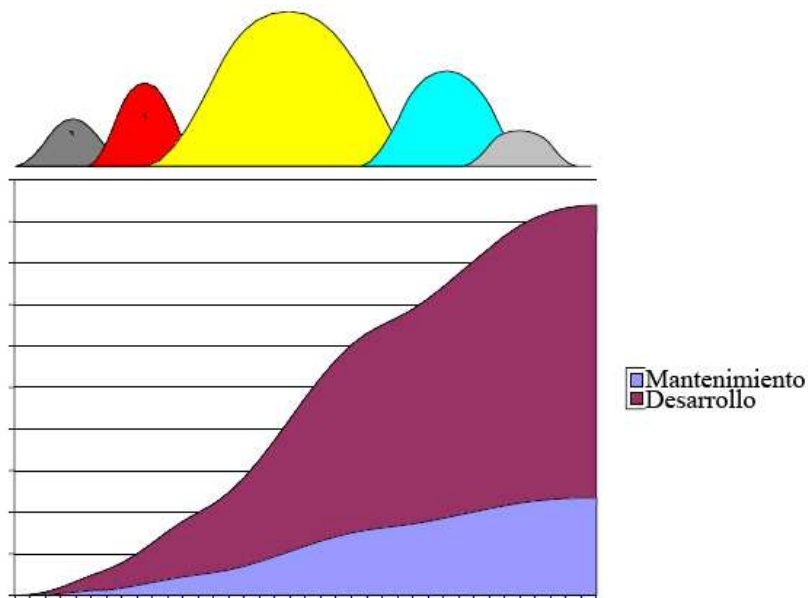
El ciclo de vida en cascada tiene estas fases organizadas en una secuencia: Una después de la otra sin ninguna retroalimentación aunque sólo con una interacción entre las fases adyacentes. Un error en una fase le exige revisar entre 25 % y 50 % del trabajo ya realizado en esa fase y al menos 25 % del trabajo realizado en las fases previas. Por ejemplo, si se descubre que tiene un error de diseño, está obligado a revisar el diseño global del sistema, rectificarlo en parte, y tiene que verificar si el análisis antes realizado está acorde con el nuevo diseño rectificado, a partir de un fallo de diseño frecuentemente afecta la satisfacción del requisito. Este hecho sugiere que para el ciclo de vida en cascada, esta es una realidad que el impacto de los errores en el tiempo de mantenimiento, en términos del esfuerzo a ser gastado, incrementa dramáticamente así como el tiempo transcurre. Si se considera que, separado del flujo de diseño, la parte grande de errores es descubierta durante las fases de test e implantación, puede ser seguro que el descubrimiento de errores cuesta mucho.

La Figura 1 muestra la tendencia del esfuerzo durante el desarrollo del proyecto dividido entre las diferentes fases del proyecto. Esta gráfica también muestra el esfuerzo acumulativo y el costo de mantenimiento con el paso del tiempo.

Con la finalidad de mitigar este problema, los ciclos vida iterativos, así como el espiral, ha sugerido que el proyecto sea dividido en algunas iteraciones con la finalidad de tratar de minimizar el impacto del esfuerzo del mantenimiento en el proyecto. Para tener una comparación, se ha propuesto tener un proyecto con el mismo esfuerzo acumulativo así en el ejemplo previo con la finalidad de tener un costo total que sea independiente del ciclo de vida escogido.

Debido a la división del proyecto en iteraciones, las cuales le permiten aceptar subconjuntos del proyecto que pueden ser considerados estables, el costo medio del mantenimiento disminuye con respecto del ejemplo en cascada.

Figura 1. Desarrollo del ciclo de vida (parte superior) con el relativo mantenimiento y esfuerzo del desarrollo (parte inferior) para el ciclo de vida en cascada:



En la Figura 2, los mismos parámetros para el ciclo de vida en cascada son adoptados con la finalidad de comparar gráficamente los resultados.

Repitiendo este proceso de *divide et impera* (el que divide impera), ordenando la iteración y repitiéndola frecuentemente, los principales costos de mantenimiento llegan a ser bajos y pueden ser considerados bastante constantes durante ciclo de vida del proyecto, mientras el costo total es el mismo. Esta es la hipótesis principal de ciclo de vida *Agile* y también el ciclo de vida ADPD: considera las mismas fases de un ciclo de vida clásico pero usa iteraciones muy pequeñas para ser repetidas frecuentemente.

Las iteraciones están tan cerca una de otra que las fases de análisis y diseño, pueden ser consideradas como una fase única para cada iteración, mientras las otras fases son equivalentes con respecto a otros ciclos de vida.

Otra importante diferencia entre la metodología clásica y ADPD está representada por la primera iteración, la cuál es más larga con relación a las demás y está enfocada principalmente en obtener el walking skeleton (esqueleto guía) de la aplicación. La fase walking skeleton es seguida a continuación por varias iteraciones liberadas del producto, en las cuales en cada iteración, una o más funcionalidades son agregadas. Esas fases serán detalladas a continuación, mientras en la Figura 3 el ciclo de vida ADPD utilizando la misma representación gráfica usada previamente.

Figura 2. Desarrollo del ciclo de vida (parte superior) con el relativo mantenimiento y esfuerzo del desarrollo (parte inferior) para el ciclo de vida en espiral:

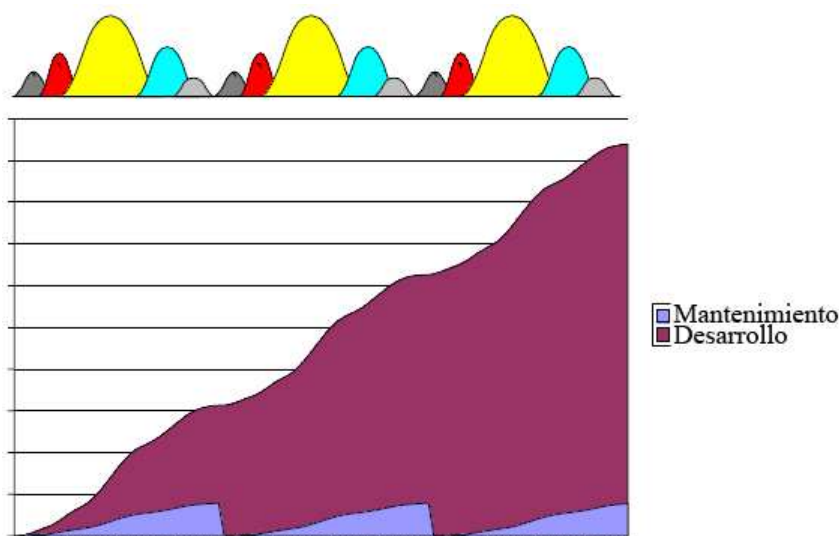
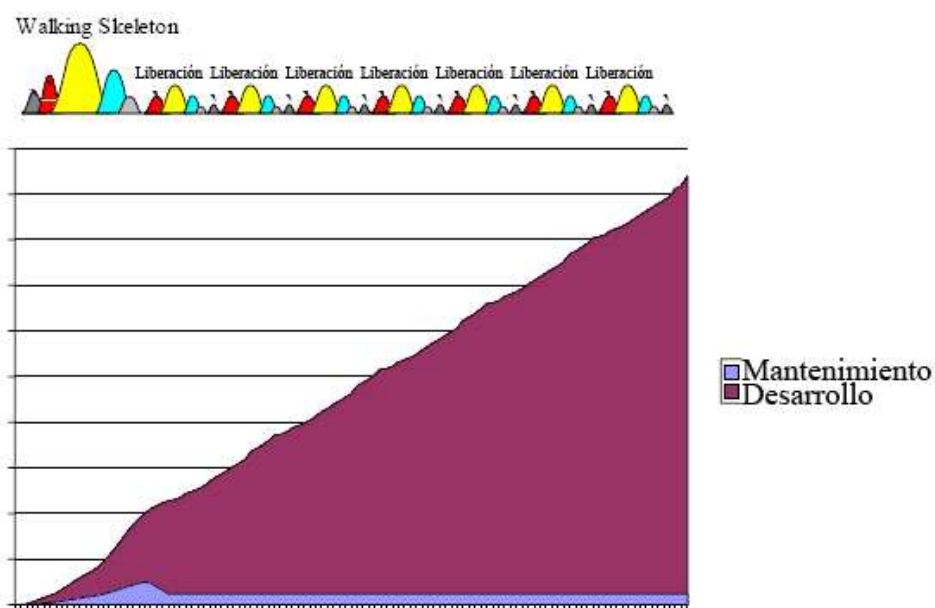


Figura 3. Desarrollo del ciclo de vida (parte superior) con el relativo mantenimiento y esfuerzo del desarrollo (parte inferior) para el ciclo de ADPD:



2.2.1 Fase Walking Skeleton (esqueleto guía)

Walking Skeleton (WS), es la primera fase del proyecto, y su mayor meta es desarrollar la arquitectura del sistema sin implementación de funcionalidad real alguna, o al menos implementar solo una ficticia o una funcionalidad pequeña con la finalidad de verificar si la estructura global del proyecto es suficientemente sólida para iniciar las iteraciones de liberación.

Durante esta fase, la diferencia entre el análisis y el diseño se mantiene. Esta iteración es normalmente grande con respecto a las otras, y es necesario tener particular cuidado en la realización de la arquitectura general.

La fase del *análisis* durante el desarrollo de WS es implementada, el diseño de casos de uso de alto nivel que guiará la selección de requerimientos generales serán llevados con el cliente. El caso de uso adoptado en esta fase es principalmente un caso de uso del negocio sin alguna referencia a la arquitectura o detalles técnicos.

Durante la fase de *diseño*, la arquitectura para tal caso de uso del negocio es decidido, y los sencillos casos de uso son detallados. La fase del diseño tiene la responsabilidad de la definición de la mayoría de las estructuras de las clases que pueden ser aplicablemente descritas por las tarjetas CRC (Classes, Responsibilities, and Collaborators).

Con la finalidad de mejorar el entendimiento de cuan importante es la fase WS y también de explicar como implementarla durante el *desarrollo*, se ha simulado la fase WS de la clásica aplicación Web de 3 capas.

Supongamos que quiere crear un site de e-commerce en la cual se tiene algunas páginas Web donde se pone una orden, confirma la canasta, y paga el total de la cantidad adquirida, una aplicación de enlace-intermedia que recibe dichos datos y conecta a una base de datos o a servicios externos. En la fase WS, se creará una página Web con sólo un par de artículos que pueden ser solicitados por el usuario, sin la necesidad de algún gráfico. El solo presentar los datos para los artículos a ser puestos en la canasta es el ID de la base de datos que identifica cada artículo. Una vez que se ha seleccionado los artículos y las cantidades a ser puestos en la canasta, se ha de pasar a otra muy simple página Web en la cual se confirma la solicitud y se selecciona el método de pago. Ésta página es realizada, eliminando todos los innecesarios adornos y sin considerar detalles reales, pero sólo pasando un flag que describe el método de pago. La capa media de la aplicación recibe esos datos y actualiza una tabla ficticia en la base de datos que contiene solo los IDs de la orden y el flag del pago. Una página de confirmación con los datos extraídos desde la base de datos es enviada al usuario.

Si todo está trabajando correctamente, se podrá ajustar algunos importantes puntos, a partir que esté seguro que:

1. La conexión entre la capa intermedia de la aplicación y la base de datos es correcta y que la base de datos esta trabajando.
2. La conexión entre la página Web y la capa intermedia de la aplicación está trabajando, y también, el servidor de la aplicación esté en marcha y corriendo.

3. La página Web trabaje correctamente, y también, el servidor Web esté trabajando correctamente y esté correctamente integrado con el servidor de aplicaciones.

4. El cliente puede ver una simulación de la celda de las transacciones después solo unos pocos días de trabajo, dando la sensación de que se está trabajando duro para ese proyecto.

La consecuencia es que la arquitectura para implementar el sistema esté listo y corriendo, y también, se puede empezar a planear con el cliente las funcionalidades para ser implementadas primero y empezar a liberar la primera versión de la aplicación, a partir al menos de que la infraestructura está aprobada.

La fase de *desarrollo* de la WS es seguida por una fase de *test y liberación*, en las cuales diferentes reglas y prácticas, descritas a continuación, tienen que ser aplicadas.

2.2.2 Fase de liberación

La fase de liberación es la fase clásica de todas las metodologías Ágile con el *análisis y diseño* de la funcionalidad seleccionada para esta fase en la primera parte, el *desarrollo y test* continúan la fase, y la *implantación y liberación* de las funcionalidades del producto planeado hasta el final. Durante todas esas fases, es necesario aplicar prácticas de ADPD con la finalidad de liberar satisfactoriamente el producto en cada fase.

2.3 PRÁCTICAS ADPD

Todas las prácticas ADPD son comentadas a continuación. Estas prácticas son divididas entre prácticas generales y gerenciales, y las reglas para ser seguidas en las diferentes fases del ciclo de vida emitidas. Es importante evidenciar que cada práctica refuerza a la otra, y en consecuencia, es fundamental adoptar todas ellas, con la finalidad de tener los beneficios ofrecidos por la metodología.

2.3.1 Prácticas de Análisis y Diseño

Durante el análisis y diseño, las mismas actividades que comúnmente son realizadas en las metodologías clásicas son ejecutadas en ADPD. La principal diferencia está en el objetivo de la fase, el cual no es más que una planificación de largo plazo pero se limita a la corta duración de la siguiente liberación. La importancia de tener cortas liberaciones muy cercanas una de otra reduce el tiempo necesitado para esas fases, aunque algunas actividades formales e importantes tienen que ser realizadas.

2.3.1.1 Pequeñas liberaciones

La importancia de tener pequeñas liberaciones ya ha sido evidenciada desde el punto de vista del costo del mantenimiento que es reducido y puede ser considerado constante durante la evolución total del proyecto. Otro importante aspecto que ha sido considerado es relativo al hecho que es simple definir para el cliente un objetivo a corto plazo con actividades bien definidas que las actividades a largo plazo que posiblemente el cliente necesitará cambiar considerablemente en el futuro.

El concepto de pequeñas liberaciones es enfocado en selección de las actividades más importantes a ser completadas a corto plazo. La ventaja de este acercamiento es que usualmente, las actividades más importantes en el corto plazo son las funcionalidades que el cliente puede definir con un mínimo riesgo de errores, a partir de que han sido enfocadas y discutidas a fondo.

El concepto que también es usado en ADPD es la suposición siguiente: si no se puede planear correctamente, al menos ha de planearse frecuentemente. Se estará respaldado en este proceso por la adopción de pequeñas liberaciones.

Esta práctica es obligatoria para una satisfactoria adopción de ADPD.

2.3.1.2 Cliente en el sitio

La presencia del cliente durante el desarrollo del proyecto es un importante valor agregado para el proyecto, la presencia del cliente permite la inmediata

retroalimentación en la evolución del proyecto. Además apoya a tener pequeñas liberaciones, en cuanto que el cliente esté en el sitio puede proveer sugerencias dentro de las funcionalidades a ser implementadas y además aprobar las características desarrolladas ya listas en un corto plazo.

Desafortunadamente, en un proyecto práctico y especialmente en proyectos pequeños, no es posible tener al cliente en el sitio, al menos que se traslade el grupo de desarrollo al sitio del cliente. Cuando no se puede tener al cliente en el sitio, se debería preparar un test de implantación en el sitio del cliente con la finalidad de obtener al menos la aprobación rápida de las características desarrolladas. Además, varios otros mecanismos pueden ser usados para tener un cliente virtual en el sitio: mensajes instantáneos, video conferencias, extranet por métodos de redes privadas virtuales (VPN), y otras tecnologías útiles.

También debería notarse que los documentos formales y tareas a ser producidas disminuyen si el cliente está presente, a partir de una colección frecuente de anotaciones con CRC y tarjetas de tareas es suficiente para establecer y mantener una correcta comunicación con el cliente.

Está práctica es altamente sugerida, aunque no es mandataria (si es reemplazado por un cliente virtual) para la adopción de ADPD

2.3.1.3 Planificación del evento

Los siguientes pasos pueden ser seguidos para obtener un buen resultado desde la adopción de la planificación del evento. Es evidente que con la excepción de cambios menores, la planificación del evento de ADPD es muy parecida con las reglas del similar XP.

- En cada fase-liberación, una mínima y máxima duración es impuesta con la finalidad de respetar el concepto de pequeñas liberaciones y tener predefinido el tiempo límite para las funcionalidades seleccionadas.
- Definir con el cliente las funciones (*narraciones*) que tienen que ser realizadas en la siguiente liberación. Asignar a las *narraciones* una prioridad entre uno y 100 de acuerdo con las necesidades del cliente, y también evaluar el valor y el impacto que la ejecución de la *narración* tiene en la totalidad del proyecto.

- Dividir cada *narración* identificada en partes con las siguientes características: (1) cada parte es una función bien definida; (2) cada parte es una función independiente o necesita para que este lista una función realizada; y (3) cada parte puede ser desarrollada y testeada en no más de dos o tres días por un programador o una pareja de programadores, en el caso del par de programadores. Cada parte, así en XP, es llamada *tarea*.
- Los desarrolladores y el grupo de gerentes asignan a cada tarea una prioridad que es un número entre uno y 100 y que está en el rango de más o menos de 10 con respecto a la *narración* a cuál pertenece. Esto permite implementar la tarea de una *narración* en un limitado periodo de tiempo con cada tarea cerca a la otra, pero con la posibilidad de posponer una tarea de una *narración* poca importancia para después de una tarea más importante de otra *narración*.
- Los desarrolladores seleccionan las tareas en orden de prioridad, en base a sus conocimientos, asignan para cada tarea el esfuerzo estimado en persona/días y la duración estimada en tiempo.
- Una vez que las prioridades han sido asignadas y el tiempo para completar cada tareas que han sido estimados, es posible planificar la revisión a ser ejecutada.
- Cuando la planificación de las actividades han sido completadas, la primera acción ha ser tomada es la aprobación de la planificación con el usuario, tomando particular cuidado a las prioridades de las *narraciones* y las asignaciones hechas por el grupo en base a la mayor prioridad asignada por el usuario. Se tendrán dos opciones: la planificación ajusta a tiempo la liberación o no. En el caso de que no se ajusta, se ha de discutir con el usuario cuales *narraciones* tienen que ser pospuestas par la siguiente liberación (la cual estará lista en no más de dos a cuatro semanas). Una vez que las *narraciones* son identificadas y la planificación es actualizada, los desarrolladores empiezan apropiadamente. Si el ajuste en el asignado tiempo de liberación, se puede elegir intentar finalizar la fase tempranamente o añadir algunas otras funcionalidades. En general, es mejor tratar finalizar tempranamente, a partir de una revisión de las funcionalidades ha ser añadidas puede ser una perdida de tiempo, a partir de que el cliente una vez más planea las prioridades de las nuevas funcionalidades comparado con las ya existentes.

2.3.1.4 Administración de requerimientos

En el caso de la adopción de una metodología Ágile, se tendrá que administrar requerimientos, a partir de que los requerimientos son el mayor componente de la planificación del evento. También se tiene que monitorear los requerimientos con la finalidad de seguir los cambios y las elecciones realizadas por el cliente. Este proceso puede ser hecho sin mucho esfuerzo, puede ser realizado simplemente por el llenando de una tarjeta por cada tarea. La colección y modificación de estas tarjetas permite el seguimiento de requerimientos y también la evaluación del cumplimiento del tiempo estimado para las *narraciones* y tareas.

Figura4. La tarjeta de narración y tarea para la administración ADPD de los requerimientos:

Story and Task Card - Tarjeta de narración y tareas					
Proyecto _____			Client _____		
Fecha _____	TipoActividad	Nuevo _____	Bug Fix _____	Cambio _____	Test _____
Narración/Tarea ID _____	Prioridad	Narración _____	Tarea _____		
Desarrollador(es) _____			Gerente _____		
Esfuerzo Estimado (horas/días) _____				Fecha: _____	
Esfuerzo Real (horas/días) _____				Fecha: _____	
Descripción de la tarea					
Notas					
Seguimiento					
Fecha	Liberación	Comentario			

La típica tarjeta de tareas es reportada en la Figura 4. Este es sólo un ejemplo del cómo una tarjeta de tareas puede ser realizada en papel, pero de seguro, puede ser una plantilla electrónica, o mejor aún, una plantilla Web, la cual permite una planificación automatizada de las actividades en concordancia a las prioridades y una verificación visual de un estimado vs el esfuerzo real gastado. En este caso, se tiene que gastar algún tiempo desarrollando el sistema de automatización, pero puede rehusarlo en todos los futuros proyectos.

Un sistema para colección de tarjetas de tiempo es una ayuda válida para la creación de modelos para la estimación de costos, es una importante fuente de información relacionada con el esfuerzo gastado en cada tarea.

2.3.2 Prácticas de desarrollo

Las actividades que ya han sido discutidas no están relacionadas al principal alcance del desarrollo-programación del proyecto. En esta sección y la siguiente, las prácticas de ADPD relacionadas a la programación y el testeo serán presentadas y comentadas. Estas secciones enfatizarán las prácticas de desarrollo, mientras que en la última sección se encontrará las prácticas de gerencia que están fuertemente relacionadas al desarrollo.

2.3.2.1 Programación en par

La programación en par es uno de las fundamentales actividades en cada proyecto XP. Este no está fuertemente enfatizado en ADPD, aunque esta adopción puede ser recomendada. La programación en par es un valor añadido cuando es posible implementarlo, pero este no es un requerimiento; al contrario, se debe evitar usar para una metodología Ágile tal como ADPD en algunos entornos que no están listos o simplemente son demasiado pequeños que no permite la aplicación de la programación en par.

ADPD impulsa el uso de la programación en par sin decir que esta es una práctica obligatoria. En cualquier caso, se analiza las ventajas de las prácticas de desarrollo. Una de las ventajas de la programación en par es la capacitación dentro del trabajo el compartir conocimientos y la habilidad para reducir la deficiencia de los desarrolladores que son nuevos en el proyecto o para una tecnología adoptada en el proyecto.

En cualquier caso, cuando es imposible aplicar la programación en par, ADPD sugiere un método alternativo para compartir conocimiento: los 15 minutos diarios de reunión que está detallada en la sección de gerencia.

2.3.2.2 Estándares de codificación

Durante el desarrollo, la adopción de estándares de codificación es una práctica obligatoria con la finalidad de evitar pérdida de tiempo tratando de interpretar cómo otro desarrollador pone seguros en el código. Ahorra este tiempo, y usarlo para mejorar el código del colega, cuando sea posible.

Estándares de codificación no solo son paréntesis y sangrías sino que también puede ser aplicado a comentarios y nombres de variables.

Los comentarios deben estar escritos con la finalidad de generar documentación automática y debe estar puesta en un lugar estándar. (p.e., siempre antes del código para el cual el comentario se relaciona) o escrito con una notación estándar, como en el siguiente ejemplo:

```
void myfunc()  
{  
// Inicio – Nombre del desarrollador – fecha  
... aquí va el código del comentario  
// Fin – Nombre del desarrollador – fecha  
}
```

Un nombre de variable y constante debe seguir una notación uniforme a lo largo de la codificación. Las constantes siempre pueden ser escritas en letras mayúsculas y las variables en Notación De Camello (p.e., estaEsElNombreDeLaVariable), con la primera letra mayúscula (p.e., Estaeselnombredemivariable), o sin letras mayúsculas pero con guión bajo (p.e., esta_es_el_nombre_de_mi_variable). Los nombres de las variables no deberían ser demasiado genéricos, pero estos deben expresar lo que ellos contienen. Para lenguajes no tipados, así como PHP, y también para los que son tipados, es posible anteponer a la variable con una corta descripción del tipo, de manera que se pueda distinguir el tipo a primera vista (p.e., intMiVariable, strMiVariable, donde el primero es un integer, y el último es un string).

Se puede perder algunos segundos tipando, pero se ganará bastante tiempo leyendo el código de otro desarrollador o leyendo el código escrito tres meses atrás.

2.3.2.3 Adopción del Sistema de Versión Concurrente (CVS)

La adopción de un repositorio CVS es una práctica obligatoria en ADPD, partiendo de que garantiza varias ventajas durante el desarrollo de software, Algunas de esas ventajas son listadas y discutidas a continuación:

* Los cambios concurrentes son permitidos. Esto significa que el *depósito* de un archivo manipulado por más de un usuario sea administrado y permita al último usuario quien operó el *depósito* decida cuál modificación tiene que ser almacenada. Esto previene la sobre escritura no autorizada de archivos sin un permiso, cuando más de una persona está trabajando en el mismo pedazo de código. Esto no es aplicable para diferente gente trabajando en el mismo código durante la fase de liberación, pero puede ocurrir, especialmente durante la fase WS, a partir de que no hay mucho código para trabajar.

* Fácil localización. Otra ventaja de un sistema de control de revisión genérica es la fácil localización de una modificación y la posibilidad de ir para atrás hacia previas versiones.

* Registro de cambios. El correcto uso de los CVS permite automatizar la escritura del registro de cambios mediante el registro del *depósito* que cada desarrollador tiene que escribir. La colección de estos registros, cuando son bien escritos (y ellos deben estar bien escritos), generan el historial documentado del código. Hablando de acuerdo con la declaración de Ágile, si existe valor en el código, nosotros tendremos más de un repositorio CVS.

2.3.3 Prácticas de test

Ésta sección describe la metodología para testear que debe ser adoptada durante el desarrollo del código. Las dos grandes categorías de test que existen en ADPD son las mismas con respecto a XP: Test unitario y test funcional. Es importante entender que cada categoría de test representa e indica dónde esta debe ser aplicada.

2.3.3.1 Test unitario

Un test unitario es la base para implementar el testeo de regresión definido en la sección de implantación, la cual permite verificar el impacto de una modificación en el previo código desarrollado. La principal receta para implementar una correcta práctica del test unitario es seguir la estrategia, codifica un poco, testea un poco, codifica un poco, testea un poco. Esto no significa que un test unitario es un test pequeño sino que tiene que testear cada función que se codifique. Los tests unitarios son escritos por el programador, pero esta práctica también impacta profundamente a la gerencia, permitiendo mejor control del proyecto. Dado un ejemplo de test unitario relacionado a la orientación a objetos. Se sugiere que escriba al menos un test unitario por cada clase. El test unitario de clase puede ser creado incrementalmente así como se agreguen funcionalidades a las clases. Suponiendo que se tiene que crear una clase que tiene un constructor y un par de métodos funcionales; una vez que se tiene codificado el constructor, se puede crear la primera parte del test unitario que, por ejemplo, realiza las siguientes tareas:

- * Crear una instancia de la clase con el constructor y verificar que los parámetros de inicialización han sido colectados y almacenados correctamente;
- * Crear dos instancias de la clase con los mismos parámetros de inicialización, y verificar que los objetos resultantes sean los mismos; y
- * Crear dos instancias de la clase con diferentes parámetros de inicialización, y verificar que los objetos resultantes sean diferentes.

Se tiene simulado la posible condición de error del constructor. Ahora, se puede añadir un método (codificar un poco) y añadir al test unitario una o más líneas de código para verificar el comportamiento del método (testeo un poco). Cada programador debería gastar alrededor del 25% de su tiempo escribiendo tests.

Mostraremos un ejemplo escrito en un meta-lenguaje orientado a objetos relacionado a una clase que implementa un número complejo:

```

Class ComplexNumber {
real: float
img: float
Constructor(float real, float img): void
Getreal(): float --retorna la parte real
Getimg(): float – retorna la parte imaginaria
}

```

El test unitario puede ser una clase (como de costumbre hecho en Java o cuando un test de suite es presentado) o un método que puede ser condicionalmente no compilado para liberación (como muchas veces ocurre en Small Talk o C++). En cualquier situación, el test unitario es algo como el siguiente código (donde Assert es una función que existe con error donde la condición no es satisfactoria).

```

Unit Test {
a: ComplexNumber(10,5);
Assert(a.Getreal()==10 AND a.Getimg()==5);
b: ComplexNumber(10,5)
c: ComplexNumber(11,6)
Assert(a==b)
Assert(a<>c)
}

```

En la clase del número complejo, si se requiere añadir la funcionalidad de la suma de manera que el método sum que realiza la adición entre la actual instancia y el objeto recibido como argumento.

```

Sum(ComplexNumber b) {
-- codigo va aquí --
}

```

El test unitario puede ser integrado por una línea que testea si la suma es ejecutada correctamente.

```
Unit Test {  
-- previous code here --  
d: ComplexNumber(1,1)  
Assert(c==b.sum(d))  
}
```

El ejemplo muestra cómo la actualizar incrementalmente un test unitario de una clase. Para grandes clases, algunas veces es necesario implementar más de un test unitario.

2.3.3.2 Test de Suite (Test Suite)

El perfecto complemento para el testeo unitario en una suite para automatizar la ejecución del test unitario y automáticamente coleccionar los resultados en un formato estándar. Si no está disponible el uso de una suite que ya existe (<http://www.xprogramming.com/software.htm>) debería planificarse alguna para el grupo. Una simplificación de un test de suite que puede ser aplicada provechosamente consiste en un directorio en el repositorio CVS donde se pone todos los archivos que ejecutan un test unitario. Antes de *depositarlo*, se puede manualmente o automáticamente ejecutar todos los tests con un script que colecciona los resultados, y si todos los tests pasan satisfactoriamente, se autoriza el *depósito*.

De seguro, la integración del test de suite en el entorno de desarrollo, si se usa alguno, es preferible. Uno de los más usados para java es el JUnit suite, el cuál está integrado en algunos editores y también puede ser usado con editores comerciales y suites.

2.3.3.3 Test Funcional

Mientras el test unitario es creado, implementado, y ejecutado por desarrolladores para verificación de código con la finalidad de crear un entorno aplicable para un testeo regresivo y de bajo costo de mantenimiento, el test funcional es creado y algunas veces implementado por el cliente con la ayuda del grupo para las partes que implica escribir código. El concepto de un test funcional es muy importante, este soporta el concepto de pequeñas liberaciones.

Cuando una funcionalidad es seleccionada para la implementación en la liberación, es necesario definir el test funcional que tiene que ser satisfecho por la funcionalidad correctamente implementada.

Es también importante notar que cuando el test funcional trabaja (así como el test unitario), significa que la funcionalidad puede ser considerada estable. Esto no significa que la funcionalidad no será modificada en el futuro, una continua mejora es uno de los principales resultados que pueden ser obtenidos con Agile programación, sólo que el código escrito esté trabajando correctamente y de acuerdo a la especificación del usuario. También significa que si no hay cambios en el requerimiento, esa función está completada, el código puede ser considerado un punto estable.

El test funcional puede ser integrado o no como el test de suite previamente mencionado. En general ellos están separados, si los requerimientos cambian, el test funcional puede ser profundamente modificado, y a nivel general, ellos pueden ser considerados como una alta capa de abstracción, siempre que ellos sean estrictamente atados a un código.

2.3.4 Prácticas de implantación y liberación

2.3.4.1 Test de regresión

Todas las prácticas que han sido previamente discutidas son enfocadas en obtener un robusto código que implementa la funcionalidad seleccionada por el usuario sin cambiar el comportamiento de las funcionalidades previamente trabajadas. Esto es posible para una correcta implementación de los tests unitarios, el cual permite la aplicación de la metodología del test de regresión.

De acuerdo a webopedia.com (www.webopedia.com), la definición del testeo de regresión es el siguiente:

El selectivo retesteo de un sistema de software que ha sido modificado para asegurar que cualquiera de los fallos han sido corregidos y que no hay funciones previamente trabajadas que fallen como un resultado de las reparaciones y que

características recién añadidas no han creado problemas con previas versiones del software. También es llamado como testeo de verificación, el testeo de regresión es iniciado después de que un programador ha logrado solucionar un problema reconocido o ha añadido código fuente a un programa que puede haber introducido errores inadvertidamente. Es una medida de control de calidad para asegurar que el código modificado recientemente aún cumple con los requerimientos especificados y que el código no modificado no ha sido afectado por la actividad de mantenimiento. (http://www.webopedia.com/TERM/r/regression_testing.html).

Como evidencia en esta definición, el test de regresión y, por lo tanto, el test unitario incrementan la calidad del código identificando ocultos o nuevos errores introducidos recientemente. La estrategia de desarrollo de ADPD garantiza el testeo de regresión, la cual es reforzada por el hecho de que el mantenimiento es el estado en el cual el código está en cada punto del proyecto, independiente de la fase de ciclo de vida. Por lo tanto, siempre el mantenimiento del código debe ser controlado de acuerdo al proceso definido por el control de calidad. El testeo de regresión es uno de los mayormente aplicables.

2.3.5 PRACTICAS GENERALES Y DE DIRECCION

2.3.5.1 Documentación del Proyecto (PD)

Como es evidente en el desarrollo Agile, es mejor tener un código trabajando con respecto a la documentación comprendida. Sin embargo, no se tiene documentación hasta que todo no sea una buena práctica. Hay dos tipos diferentes de documentación en el proyecto: la documentación generada automáticamente y la documentación denominada proyecto ADPD, referida como PD en adelante. Mientras que el primer tipo de documentación ha sido descrita en la sección relacionada al desarrollo, la más reciente será detallada en esta sección, es una de las principales deberes del gerente del proyecto. Hay dos diferente ventajas para realizar un PD: (1) realizar un documento clásico como reporte del seguimiento que puede ajustarse a la necesidad para grandes organizaciones; y (2) una manera más Agile de compilar documentación con el uso de herramientas para generación de código automático iniciando desde los

diagramas de UML. Esta segunda ventaja será detallada más adelante (Adopción de ADPD y UML), la cual discutiremos la relación entre UML y ADPD, mientras en los párrafos siguientes, una corta descripción de la manera clásica para documentar el proyecto es presentada junto con una posible tabla de contenidos del PD.

El esqueleto de PD es definida en la fase WS. En general, este documento está a continuación escrito siguiendo las líneas directivas del plan de calidad desarrollado por el Departamento de Garantía de Calidad (Quality Assurance Department), pero en cualquier caso, éste debe contener una corta descripción de la visión del proyecto y objetivo, y una descripción general de la arquitectura a ser usada e implementada en la fase WS. Después de esta sección introductoria, una sección en la cual todas las tarjetas de tareas son recolectadas, es necesario con la finalidad de coleccionar el historial de cada tarea, las estadísticas, y los cambios realizados. La última sección del documento debe contener una lista de todas las liberaciones del producto con una descripción corta de todas las funcionalidades implementadas. Es también necesario reportar la medición realizada y los correctivos relevantes que han sido aprobadas durante las reuniones diarias.

Una base para construir el resumen del PD, junto con la longitud máxima sugerida para cada sección, puede ser extraída desde el listado siguiente, tomando en cuenta que PD debe ser funcional para la visión del proyecto, pero que también debe estar de acuerdo a las especificaciones del Departamento de Calidad.

1. Antecedentes del Proyecto e Introducción. 2 páginas.
2. Visión y Objetivo del proyecto. 2 páginas.
3. Descripción de la Arquitectura. 4 páginas.
4. Historial del Proyecto.
 - 4.1. Tarea 1
 - 4.1.1. Tarjeta de tareas y narración. Depende del número de narraciones y la complejidad de las tareas.
 - 4.1.2. Decisión 1, tarea relevante. 1 página.
 - 4.1.3.
 - 4.1.4. Decisión n, tarea relevante. 1 página.
 - 4.2. ...
 - 4.3. Tarea m
 - 4.3.1. ...
5. Medición del proyecto. 2 páginas con la descripción de las métricas adoptadas.
 - 5.1 Seguimiento del esfuerzo. Depende del tamaño del proyecto.
 - 5.2 Métricas del proyecto. 3 páginas
 - 5.3 Otras Métricas. Depende de las métricas adoptadas.

2.3.5.2 Agile Intergroup Improvement Team (AIIT)

El Agile Intergroup Improvement Team (AIIT) es tarea de ingeniería en la cual es mandatario incluir todos los grupos de gerentes de los diferentes grupos de la compañía y el gerente senior de la compañía, con particular referencia al Gerente Técnico.

AIIT es importante y necesario donde la compañía es bastante grande por tener más de un grupo trabajando en diferentes proyectos. AIIT es un grupo de ingeniería que trabaja a nivel de la compañía que tiene la finalidad de cumplir con las necesidades organizacionales que no son consideradas en XP y tampoco en otras metodologías Agile.

Las actividades principales de AIIT son relacionadas a decisiones que pueden influenciar la estrategia de la totalidad de la compañía en términos de herramientas estándar y metodologías a ser empleadas y en términos de capacitación planificada.

La necesidad de AIIT nace desde el hecho que, durante el desarrollo de proyectos en una compañía, es comúnmente asumido por un grupo de ingeniería que se reúne periódicamente para establecer estrategias, ocupándose de problemas de control de calidad, definiendo las ventajas de metodologías para la compañía, y acarreando otras actividades. Este equipo es requerido por el clásico aprovechamiento del software de calidad, así como el Modelo de Madurez de Capacidad del Software, y también para los procesos de control de calidad el ISO 9000.

Con esta finalidad, AIIT tiene que hacer lo siguiente:

* Revisar las ventajas de la metodología con la finalidad de verificar si estas pueden ser aplicadas a todos los equipos en la compañía. La adopción de la programación en par o no, decidir la duración de las reuniones diarias, y así sucesivamente.

* Revisar y aprobar las herramientas que uno de los equipos ha adoptado con éxito, obtener ventajas en productividad, mejorar la calidad del código, o mejor usabilidad de las prácticas. Si algunas de estas ventajas son demostradas, AIIT puede extender la adopción de estas herramientas para los otros equipos que necesiten tales ventajas.

* El plan de capacitación a nivel de la compañía con la finalidad de reducir la falta de experiencia de los desarrolladores. La capacitación puede ser en el trabajo, realizado por gente de la compañía, o realizado por compañías externas mediante subcontratos.

* Las actividades del plan de capacitación para gerentes (p.e. equipo de gerentes) con la finalidad de mejorar el enfoque de las actividades que serán realizadas, para guiar al equipo, y enfocar al equipo de gerentes en cómo obtener mejores resultados en el caso de resultados pobres, metas no realizadas, mala calidad del código, o mala retroalimentación desde un cliente. Ésta capacitación es usualmente realizada por una compañía externa, incluso se lo puede hacer por equipos expertos internos de gerentes.

2.3.5.3 Seguimiento y Medida

Es importante seguir la evolución del proyecto en términos de asignación de tareas y roles con la finalidad de identificar responsabilidades. Las metodologías Agile, en general, no tienen mucho cuidado en cuanto los gerentes; la mayor desventaja, si la gente no es responsable de sus acciones, es la pérdida total de control sobre el proyecto. Es necesario establecer mecanismos de seguimiento que sean útiles, no solo para identificar quien ha hecho algo errado, sino también para conocer quien de entre los desarrolladores ha modificado y, por lo tanto, conocer detalles de una parte especificada del código. Esta actividad de seguimiento también es útil cuando una persona decide abandonar el equipo o la compañía, se puede identificar en un corto tiempo quien fue compartido del conocimiento de una funcionalidad particular e identificar cuantas funcionalidades han sido desarrolladas sólo por esa persona. También se puede fácilmente identificar la cantidad del tiempo de capacitación que se necesitará con la finalidad de transferir el conocimiento a otras personas.

La actividad del seguimiento puede ser lograda aplicando la tarjeta de tarea y narración en la cual esta la constancia de cada tarea/narración desarrollada y cual fue el gerente para cada funcionalidad.

Otro importante dato a ser seguido, medido, y analizado son los esfuerzos planeados antes de que el proyecto empiece por publicistas/vendedores o el gerente senior, una *a priori* estimada por desarrolladores durante la planificación del evento, y una evaluación *posteriori* cuando la tarea/narración ha sido completada.

Esta aproximación de la medición del esfuerzo tiene tres grandes ventajas:

- * Entrega la retroalimentación al gerente senior y publicistas/vendedores acerca de la estimación del esfuerzo durante la negociación con un cliente en relación a la capacidad del equipo que tiene en dominar el proyecto.
- * Entender desde la estimación y evaluación subsiguiente cómo evaluar correctamente el esfuerzo para cada microfase del proyecto. Esto incrementa la capacidad de los desarrolladores para estimar la productividad de ellos en base a las tareas a ser realizadas.
- * Evaluar la productividad del equipo y tener una clara imagen de las tareas que retrasarán la planificación. Si la colección del esfuerzo y/o la tarjeta de tarea escrita es automatizada, también puede utilizarse una herramienta que automáticamente entregue retroalimentación de la planificación. Esta es una manera para que el gerente senior para evaluar la capacidad del equipo de gerentes con respecto a las metas planeadas.

La medición del esfuerzo no es la única manera de realizar la medición del proyecto. Algunos otros aspectos del proyecto pueden ser medidos y evaluados para tener una retroalimentación acerca de la evolución y calidad del proyecto. Aparte de las métricas para la estimación de esfuerzo que pueden ser aplicadas o no, hay algunas otras métricas de proyecto para evaluar gerentes, productividad y aplicabilidad de la metodología adoptada son presentadas, junto con métricas para sistemas orientados a objetos y métricas cognitivas con la finalidad de medir la comprensibilidad del código y, por lo tanto, el proyecto.

2.3.5.4 Quince minutos de Reunión permanente Diaria

Los 15 minutos de reunión permanente diaria pueden ser un buen reemplazo para la programación en par en todas las situaciones en las cuales no pueda ser aplicada. Es necesario especificar los dos principales aspectos de esta práctica: por qué diariamente y por qué 15 minutos permanentes.

* Reunión Diaria. Es importante compartir conocimientos cada día, porque las actividades diarias, selección de tareas relacionadas, problemas, y soluciones han de ser comunicadas a los otros miembros. Esto permite presentar a los otros miembros del equipo que se ha hecho durante los días previos. La frecuencia de la reunión es importante con la finalidad que sea segura la transferencia de todas las cosas y transferirlo en un período corto de tiempo.

* Quince Minutos Permanentes. Una reunión con una duración de 15 minutos parece ser demasiado breve para comunicar algo a alguien, pero si la hipótesis previa (reunión diaria) es completada entonces, 15 minutos pueden ser suficientes para revisar un día estándar de desarrollo. Considerando el tiempo total de las reuniones en un mes estándar (21 días), es más de 5 horas de reunión, y por lo tanto, puede ser equivalente al menos un medio día de reunión por cada mes. La ventaja de tener 15 minutos cada día es que el enfoque de la reunión está en un pequeño número de problemas que son rápidamente tratados. El concepto de una reunión permanente es también importante, si se está forzado a permanecer en frente de un pizarrón, no hay inclinación por pérdida del tiempo, y por lo tanto, hay una alta probabilidad de que el tiempo planificado diario sea suficiente.

Si, durante la reunión, algunos problemas surgen, estos podrían necesitar una reunión restringida con los desarrolladores involucrados con el problema con la finalidad de buscar una solución aplicable. La solución identificada también será comunicada a los otros miembros en la siguiente reunión diaria. Con esta aproximación, sólo un limitado número de personas tiene que tratar con una reunión más larga que los 15 minutos, mientras los otros pueden inmediatamente iniciar a codificar o realizar las otras actividades diarias.

2.3.5.5 Capacitación

La capacitación es una actividad muy importante para el rendimiento y debe estar enfocado tanto en desarrolladores (para reducir la falta de experiencia o para introducir una nueva tecnología o lenguaje) y en el equipo de gerentes con la finalidad de enfocar el esfuerzo de gerencia en obtener los mejores resultados desde la metodología y la mejor integración en el entorno de la compañía y calidad del sistema.

La capacitación es planeada a nivel de la compañía y no a nivel de equipo durante reuniones AIIT con la finalidad de resolver problemas que pueden surgir a nivel de equipo.

2.4 Adopción de ADPD y UML

En el mundo de la gerencia y desarrollo Agile siempre sugiere que se adopte la tecnología más simple y la metodología que puede ajustarse a los proyectos con los que se negocia.

Existen inquietudes que pueden surgir: Cómo puede UML ajustarse a estos requerimientos?. Puede UML ser empleado con metodologías Agile y particularmente con ADPD?. Cuantos tipos de diagramas deberían ser usados apropiadamente con ADPD?. Este capítulo tratará de responder éstas y otras interrogantes que son formuladas al considerar a UML como herramienta para respaldar el desarrollo y la gerencia de proyectos Agile. Los desarrolladores Agile usualmente dibujan diagramas UML en un tablero con la finalidad de tener un tema para discutir acerca del formalizado bien conocido lenguaje común. De otra manera, cuando esos diagramas son transferidos a una herramienta capaz de manejar la notación UML, se obtiene la ventaja de tener una documentación generada automáticamente que puede ser actualizada para lo que se necesite.

UML y la documentación automatizada que algunas herramientas pueden generar, parten desde un paquete de diagramas UML, puede sustituir la Documentación del Proyecto ADPD. Esta ventaja en la documentación implica al

gerente del proyecto a hacer algunas elecciones y decide si algunas partes del proyecto han de ser documentadas o no. En cualquier caso, es mejor que esta elección sea realizada por el gerente del proyecto, quien puede seleccionar las más importantes características a ser documentadas de entre la totalidad del proyecto en vez de obligar al lector de la documentación para reducir de acuerdo a su juicio.

2.5 UML en breve

Ésta sección no es un exhaustivo ensayo en diagramas y uso de UML no es una enfocada introducción a los conceptos generales de UML. El principal propósito de esta sección es proponer la vista del autor de UML y la adopción UML, para la lectura adicional para los lectores interesados, y para dar una breve introducción a las herramientas que pueden ser útilmente empleadas para aprovechar UML por primera vez.

UML es el acrónimo de Lenguaje de Modelamiento Unificado y, como el nombre sugiere, puede ser útilmente empleado para modelar con un lenguaje (visual) todos los procesos. UML trata el unificar para la adopción de un paquete de diagramas de todas las posibles opciones para modelar los procesos del negocio, interactuando entre modelos de software, y la descripción humana de jerarquía de clase o dependencia de módulos junto con la breve descripción de módulos o la funcionalidad de clases y responsabilidades.

Cuando esta sección fue escrita, UML estaba en la especificación 2.0, por lo tanto, puede ser considerada suficientemente madura para ser usada.

Con la finalidad de dar al lector la vista más clara de los diagramas y presentar algunas herramientas gratuitas (o las versiones descargables gratuitas de herramientas comerciales) para adopción de UML y desarrollo en Java, el autor ha decidido adoptar dos herramientas gratuitas que pueden ser descargadas desde Internet: NetBeans 3.6 IDE para Java (www.netbeans.org) y Smart Development Environment Versión 1.1 de Visual Paradigm en su versión para la comunidad para NetBeans y SunOne (www.visual-paradigm.com). Las

instantáneas de esta sección han sido obtenidas durante el uso de dichas herramientas. La herramienta usada en este capítulo ha implementado la versión 1.4, y el estándar de diagramas UML que son propuestos son descritos a continuación:

* Diagrama de Casos de Uso. El diagrama de casos de uso describe que hace sistema desde el punto de vista de un observador que es externo al sistema. El enfoque del diagrama de casos de uso es describir que tiene que hacer el sistema sin especificar cómo el sistema hace dichas funciones. El diagrama de casos de uso está cercanamente conectado a *metáforas* de XP o a *narración* de Agile, los casos de uso describen que ocurre cuando un actor (p.e. una persona u otro sistema) interactúa con el sistema según la descripción.

* Diagrama de Clases. La finalidad de un diagrama de clases es mostrar las clases pertenecientes al sistema y sus relaciones. Ésta sobrevista del sistema no es una vista dinámica; cuando hay interacción de dos clases es dado por el link de interacción sin descripción de lo que ocurre.

.

* Diagrama de Secuencia. Un diagrama de secuencia es un diagrama dinámico que muestra que ocurre durante un tiempo. En este diagrama, todos los detalles de las operaciones son especificados, y los mensajes que cada objeto envía a otro son detallados junto con el instante de tiempo en que ocurre. El diagrama es construido con el progreso del tiempo desde arriba hacia abajo, y los objetos son ordenados de izquierda a derecha de acuerdo con el instante de tiempo en el cual ellos aparezcan en la secuencia de los mensajes.

* Diagrama de Colaboración. Los diagramas de colaboración contienen y expresan la misma información con respecto al diagrama de secuencia, pero su enfoque no está en cronometrar sino los roles del objeto. Un diagrama de colaboración es un gráfico en el cual los límites son los roles de los objetos, y los arcos son los mensajes.

* Diagrama de Estados. El diagrama de estados son los diagramas clásicos para representar las máquinas de estado. Ellos son útiles cuando es necesario definir el cambio de estado de un objeto, mientras ellos no están propuestos cuando la cooperación de objetos tiene que ser representado.

* Diagrama de Actividad. Un diagrama de actividad puede ser descrito como un diagrama de flujo con el soporte de elaboración paralela. Este puede representar los procesos del negocio y el flujo de trabajo como un simple diagrama de flujo. Originalmente, fue considerado un caso especial de diagrama de estado, pero especialmente con la versión 2.0 de UML, éste ha obtenido una identidad y rol personal en el mundo del UML.

* Diagrama de Componentes. El diagrama de componentes es adoptado para dividir el sistema en partes descritas por su interfase. La interacción entre las interfases puede ser también documentada con estos diagramas.

* Diagrama de implantación. El diagrama de implantación documenta las configuraciones físicas del software y hardware, con particular cuidado al asignar las partes del software a las correctas partes del hardware.

El paradigma visual también soporta los siguientes diagramas en el Smart Development Environment (SDE):

* Análisis textual. “El análisis textual es una de las maneras del análisis de requerimientos. Se puede averiguar y destacar los autores, los casos de uso y clases candidatos desde el planteamiento del problema. Los elementos candidatos pueden ser usados para crear modelos UML y elementos del diagrama.” (Manual de SDE).

* Flujo de trabajo del negocio. Los diagramas de flujo de trabajo del negocio son útiles para representar procesos de negocio y flujo de trabajo de documentos en procesos del negocio.

* Tarjetas CRC. “Las tarjetas CRC es otro método de análisis de requerimientos. Se puede caminar a través de los requerimientos del sistema para identificar las clases y usar las tarjetas CRC para almacenar sus nombres, responsabilidades y colaboraciones.” (Manual de SDE).

El lector que quiera más detalles acerca de UML puede referirse a Fowler (2004), Rumbaugh (1999), Booch (1998), Larman (2001), Mellor (2002), y Favre (2003). UML es un lenguaje visual que puede ser usado con papel y lápiz o en un pizarrón y que puede ser útilmente integrado con RAD e IDE para desarrollo.

2.6 UML y el Ciclo de vida ADPD

Durante el ciclo de vida ADPD, UML puede ser adoptado útilmente en diferentes fases con la finalidad de simplificar y unificar la notación usada en el equipo y para automatizar la generación de la documentación del proyecto.

2.6.1 Fase Walking Skeleton (Esqueleto Guía)

UML es adoptado principalmente para el análisis del proyecto. Especialmente para la parte relacionada al análisis del negocio de los casos de uso. Un caso de uso de negocio es muy similar a la formalización de la metáfora de XP. Colecciona lo que el sistema ha de hacer sin considerar algún detalle de arquitectura.

Suponiendo que se tiene que crear un sistema simple para una tienda e-business en el cual el usuario puede seleccionar los ítems, confirmar la canasta, seleccionar el método de pago, y finalmente comprar los ítems de la canasta, pagado por tarjeta de crédito, pagando cuando las compras lleguen a casa, y así sucesivamente. El banco está presente para autorizar el pago con tarjeta de crédito.

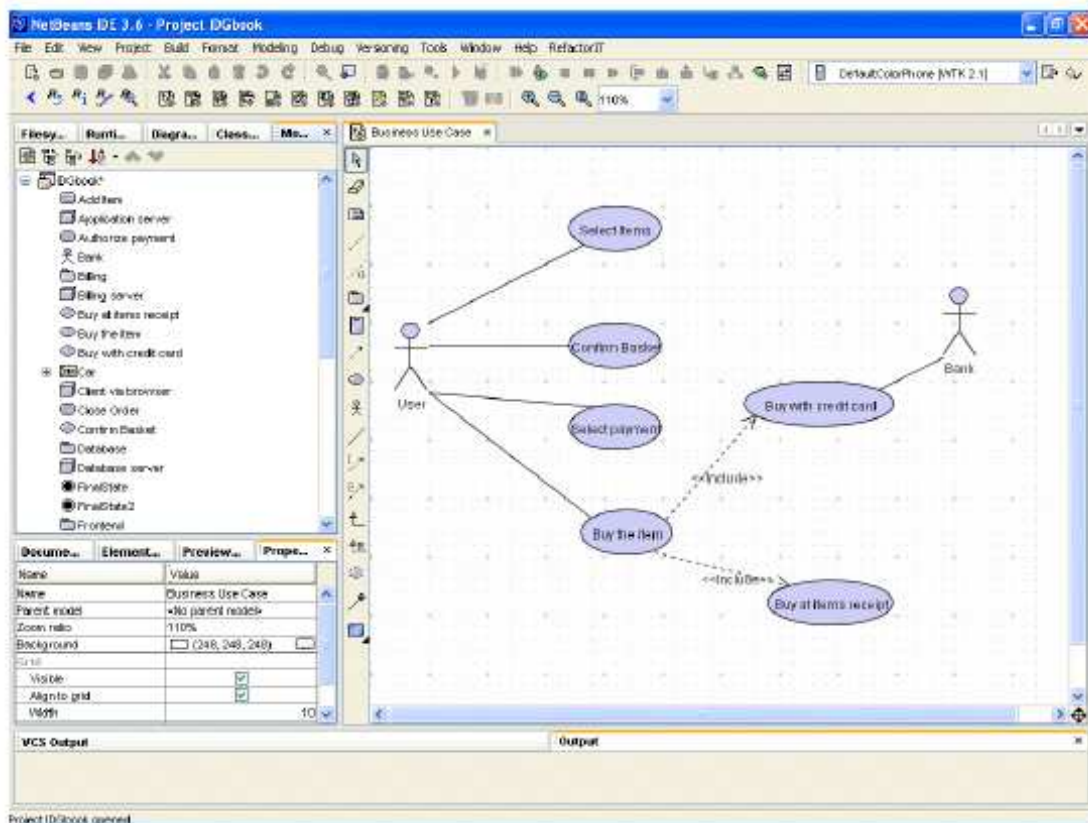
2.6.1.1 Adopción de los Casos de Uso

Los casos de uso pueden ser divididos en dos grupos principales: nivel del negocio y nivel del sistema. Los casos de uso del nivel del negocio son descripciones de alto nivel del sistema, adoptados principalmente durante la fase de Walking Skeleton para dar al cliente una idea global del sistema en el “nivel kite,” de acuerdo con la notación de Cockburn (2001a), el cual es detallado a continuación. Los casos de uso a nivel del sistema son adoptados principalmente en las fases del desarrollo, cuando es necesario tener un detalle en profundidad de la operación a ser realizada.

El supuesto sistema considerado en las previas secciones puede ser descrito por los casos de uso del nivel del negocio reportado en la figura 5. Una descripción ha sido añadida para cada elemento del diagrama con la finalidad de generar automáticamente la documentación del sistema en ambos formatos PDF y HTML.

El trabajo de dibujar y especificar algunos detalles en la descripción tiene como punto positivo tener una documentación automatizada del proyecto.

Figura 5 Diagrama de casos de uso del negocio para el sistema WEB



El ejemplo reportado es un diagrama de muy alto nivel, donde el actor bank no esta tan definido, ya que no se conoce a este nivel que clase de identidad es: humano, Web service, librería de proyecto, y así sucesivamente.

El autor piensa que los diagramas de casos de uso pueden ser muy útiles en presentar la información para clientes que no están técnicamente aptos, ellos pueden entender lo que quiere hacer a primera vista. Los casos de uso también son útiles con la finalidad de que los desarrolladores y arquitectos de software identifiquen cuales operaciones tienen que ser completadas en un sistema con la finalidad de tener el resultado deseado. Algunos expertos tal como Fowler (2004) sugiere que se use el caso de uso textual antes que los diagramas gráficos. Se inicia esta clase de casos de uso (no formalizado en UML) con la finalidad de dar al lector la posibilidad de escoger la notación que él o ella prefiera.

El caso de uso textual para el diagrama de caso de uso del negocio está resumido en la Tabla1:

Tabla 1. Caso de uso textual para compra de ítems

User buys products
Use case level: Sea Level
<ol style="list-style-type: none"> 1. User browses the Web site, selecting items to buy 2. User verifies the basket and checks out 3. Use selects payment method 4. User pays by credit card 5. Bank authorizes payment
<p>Extensions</p> <p>4a: User pays when he or she receives the items</p> <p>.1 System prepares the shipment with delayed payment and surcharge</p>

La tabla 1 colecciona más información con respecto al diagrama gráfico, y por lo tanto, es más aplicable para la gente que está más cerca del código y de la implementación del sistema. La tabla está dividida en 4 partes, detallada desde arriba hasta abajo:

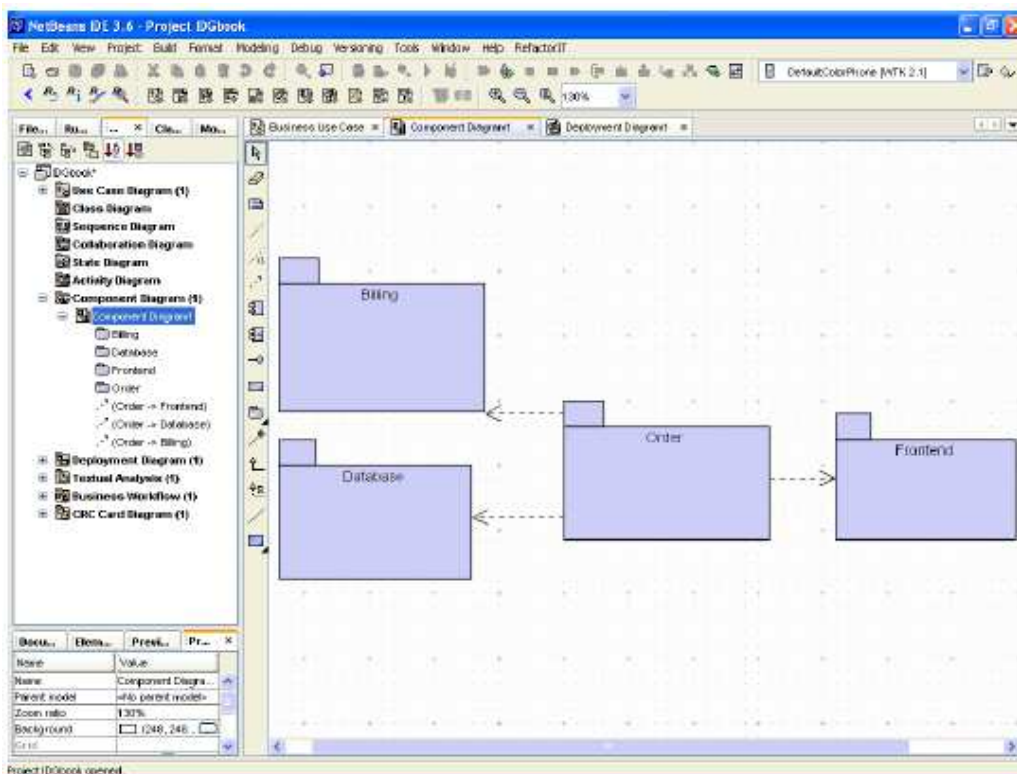
1. Título del caso de uso
2. Nivel del caso de uso. De acuerdo con la clasificación en Cockburn (2001a), los casos de uso pueden ser clasificados entre (1) sea level, cuando la acción descrita involucra mayormente el actor principal y el sistema es un área restringida; (2) fish level para todos los casos de uso que existen solo porque ellos son incluidos por un caso de uso de sea level; y (3) Kite level el cual es un caso de uso general que puede mostrar la adopción del caso de uso del sea level en un contexto más general, generalmente a nivel del negocio.
3. Descripción del caso de uso. La secuencia de actividades realizadas por un actor para alcanzar el resultado deseado (en el ejemplo, el resultado es exitoso al comprar online).
4. Descripción de la extensión. Cuando es posible tener más de una acción para un simple item dentro un caso de uso, esta sección es llenada con posibles alternativas que pueden traer a resultados objetivos o terminación con un estado de error.

2.6.1.2 Diagrama de componente (paquetes)

Una vez que se tiene definido que va a hacer el sistema, es tiempo de describir cómo el sistema está organizado a un alto nivel en términos de macro componentes. UML 2.0 ha adoptado el diagrama de paquetes que puede ser más adecuadamente empleado, pero el diagrama de componentes en la especificación 1.4 es un súper-paquete y, por lo tanto, también puede ser adoptado. El diagrama de componente que se mostrará tiene la misma semántica y apariencia de un diagrama de paquetes.

La arquitectura del sistema simple describe que previamente puede ser expresada por el diagrama de componente esbozado en la figura 6.

Figura 6. Diagrama de componente (UML 1.4) o diagrama de paquete (UML 2.0) para el sistema Web

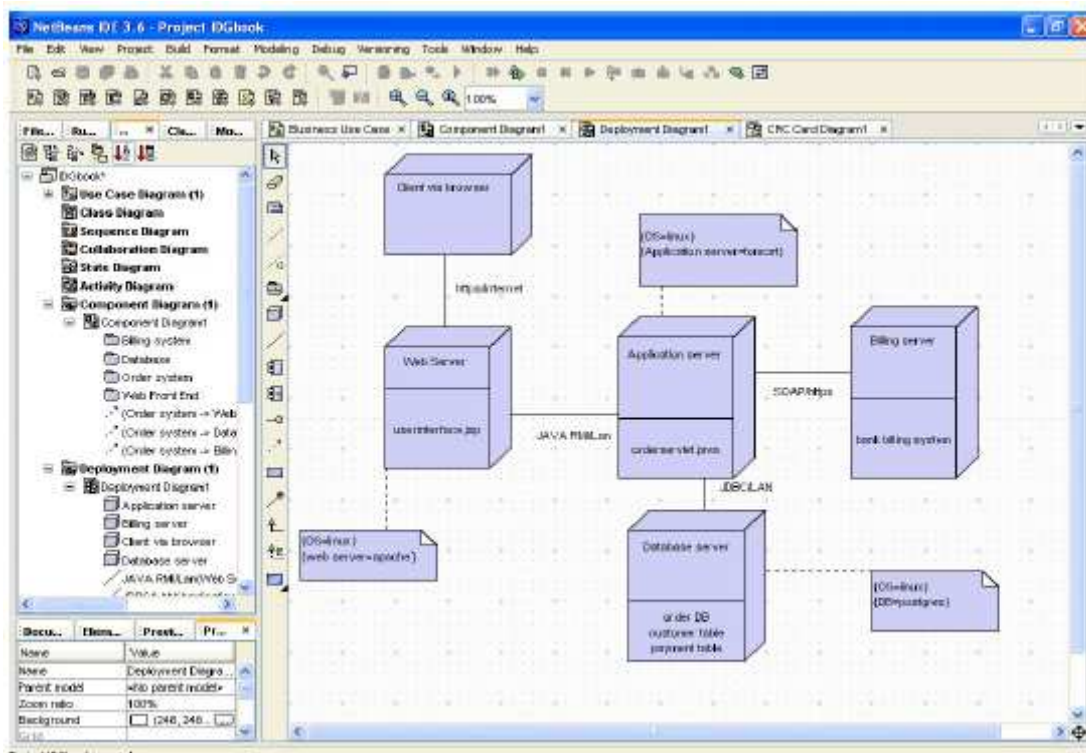


2.6.1.3 Adopción del diagrama de implantación

Con la finalidad de tener un test de planta en el sitio del cliente o en el entorno de desarrollo, es necesario tener clara la idea de cómo implantar el sistema desarrollado. UML ayuda con un diagrama que puede identificar la coincidencia entre el hardware que se ha adoptado y los módulos del software que tiene que implantarse.

La figura 7 reporta el diagrama de implantación para el simple sistema Web. Como ya se ha notado, en este caso, el mismo sistema tratado es implantado. Algunas notas han sido añadidas para algunas unidades implantadas con la finalidad de mostrar lo definido (si existe) tanto como sistema operativo, tipo DB, y así sucesivamente.

Figura 7. Diagrama de Implantación para el sistema Web



2.6.1.4 Adopción de las tarjetas CRC

Las tarjetas CRC no son un estándar UML, pero usualmente son usadas en una sesión del proyecto por los gerentes del proyecto y son frecuentemente integrados en herramientas case de UML. Las tarjetas CRC son muy útiles para

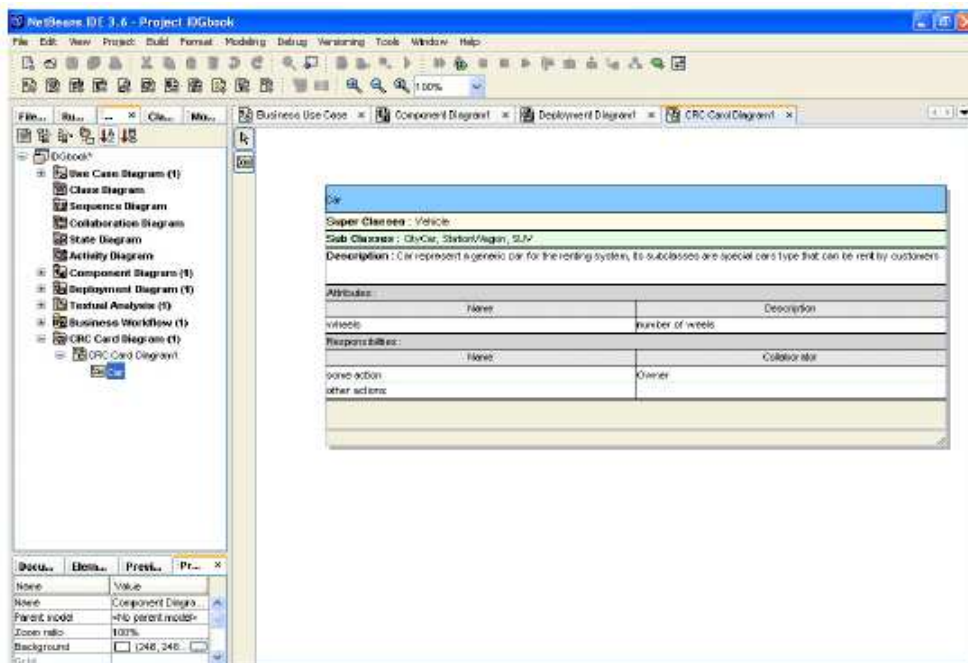
describir la responsabilidad de cada clase del sistema que se está implementando, puede ser la base para construir más diagramas de clases complejas.

Existen diferentes tipos de tarjetas CRC que pueden ser adoptados por metodologías Agile.

La más compleja y completa se ha encontrado en la herramienta SDE presentada en este capítulo, donde todas las relaciones heredadas y componentes de clase pueden ser descritos completamente.

Este primer ejemplo es muy similar a una clase de un diagrama de clase de UML pero con una baja formalidad, y puede ser adoptada como una descripción avanzada de clases, dejando al diagrama de clases el detalle de la más importante de las secciones del sistema. Un ejemplo semejante de una tarjeta CRC está en la figura 8.

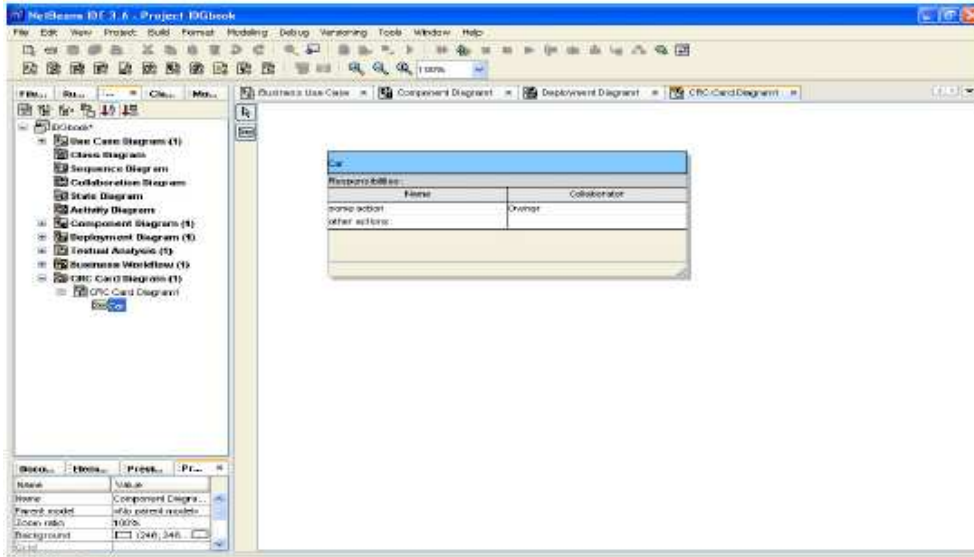
Figura 8. Tarjetas Complejas CRC consideradas por el Visual Paradigm SDE



La más simple que se ha reportado está en Fowler (2004), la cual es basada en el original trabajo de Cunningham (1989), y solo reporta la responsabilidad y los colaboradores de las clases sin algún detalle acerca de la posición en la jerarquía de clases, los atributos, o métodos, y así simultáneamente.

El segundo tipo de tarjeta CRC puede ser descrito por la herramienta SDE, así reportada en la figura 9.

Figura 9. Clásica Tarjeta CRC



Se puede adoptar todas las notaciones que son los enlaces entre esos extremos, con la finalidad de usar la más aplicable para el equipo. Se puede remover todos los detalles que se clasificaron como marginales o no útiles y almacenar solo las partes que se consideren importantes, recordando que la segunda notación es la comúnmente adoptada en la gerencia de proyectos de software. Las tarjetas CRC siempre están escritas en un pedazo de papel con la finalidad de simular el traspaso de un mensaje moviéndolo en la mesa o ascendiéndolo cuando el objeto relacionado ha sido activado. Se puede formalizar las tarjetas CRC con la adopción de una herramienta case, para simplificar su mantenimiento, en términos de modificación de la documentación que la tarjeta CRC representa, o solo para compararlo con el diagrama de clase con las que estas están relacionadas.

2.6.1.5 Adopción del diagrama de clases

Una vez que se tiene completada esta operación, se está cerca de la actividad de desarrollo de la fase walking skeleton, y es tiempo de considerar clases y sus interacciones. El diagrama de clases de UML permite al desarrollador y al gerente del proyecto detallar la descripción del proyecto al nivel deseado. Cuando se

adopta una metodología Agile, se sugiere evitar la descripción detallada. Algunos diagramas de clase que evidencian la crítica dependencia deben ser presentados cuando se tenga que comunicar a otra persona los detalles internos de implementación.

También se sugiere que se adopte el diagrama de clases en todas las situaciones en las cuales las tarjetas CRC en su más simple forma no pueden solucionar los problemas. Los diagramas de clases también pueden ser representados dentro del paquete al que ellos pertenecen en un diagrama de paquetes, pero poner más información conjunta puede ser confuso a la necesidad del lector y el punto de vista puede estar equivocado del enfoque del mensaje principal que el gerente del proyecto quisiera transferir.

Cada persona que ha visto un UML tiene un 95% de probabilidad de haberse encontrado un diagrama de clases; lo que se quiere especificar es solo un paquete de sugerencias para útilmente emplear el diagrama de clases en una metodología Agile. La principal sugerencia es adoptarla, pero con cuidado. Otras sugerencias son listadas a continuación:

- * El comportamiento del modelo estático de diagramas de clase y, por lo tanto, no puede capturar el mecanismo dinámico del traspaso de mensaje y las interacciones dinámicas.
- * El modelo no hace todas las posibles interacciones (especialmente asociaciones las cuales pueden producir frecuentes malentendidos).
- * Adoptar los estándares de lenguajes de programación en las características de implementación tal como paquetes (o namespaces), atributos, y visibilidad de métodos (especialmente protegido y visibilidad de paquete).

Dibujar diagramas para evidenciar la complejidad o para delegar el desarrollo solamente para partes críticas, y actualizarlos tan frecuente como sea posible; los obsoletos diagramas no son útiles sino pueden ser peligrosos, a partir de que ellos pueden proveer falsa entrada a los desarrolladores y arquitectos de software.

Los diagramas de clase completa la lista de los diagramas usualmente adoptados en la fase de WS. Es hora de dar algunas indicaciones relacionadas a la posibilidad de adoptar los diagramas UML y notaciones durante las fases de liberación.

2.6.2 Fase de liberación

Durante las fases de liberación, algunas notaciones UML y diagramas pueden ser empleados útilmente con los mismos objetivos reportado en la fase WS.

2.6.2.1 Adopción de los casos de uso

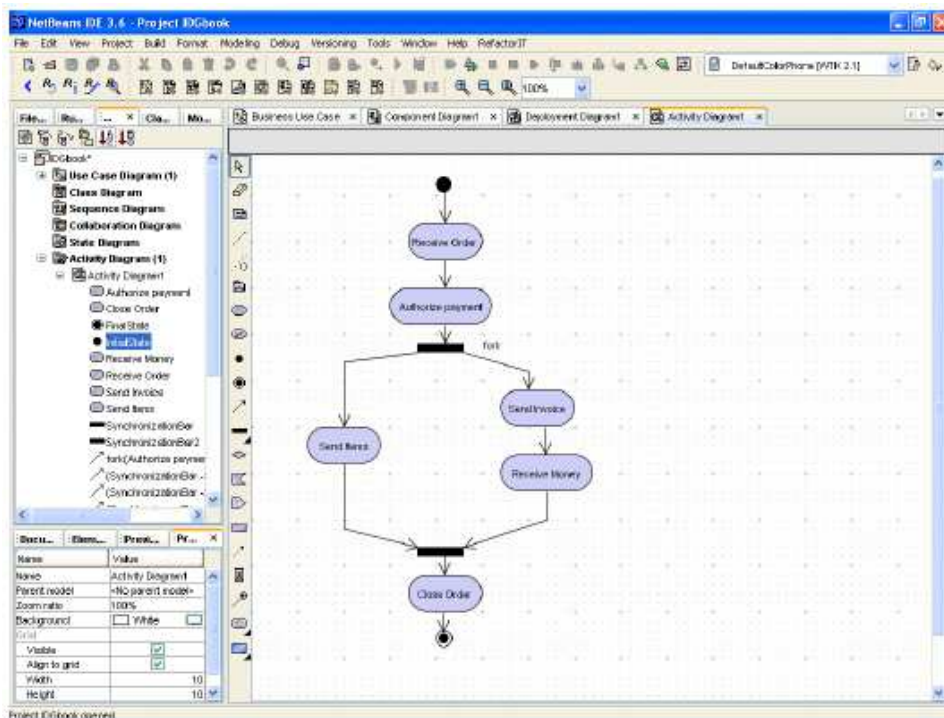
Durante las fases de liberación, puede ser útil para describir actividades particulares con un caso de uso textual para la clarificación con el cliente, quien en este punto del proyecto está involucrado profundamente en el proceso de decisión y está también más calificado con respecto al lenguaje técnico y las operaciones que tienen que ser implementadas en la *narración* seleccionada para la liberación.

2.6.2. 2 Adopción de diagramas de Actividad

Donde quiera que se tenga un objeto cuyo comportamiento es particularmente complejo o muestra un flujo paralelo de información (p.e. aplicaciones multi-hilos), se puede adoptar un diagrama de actividad que permita modelar el flujo de los objetos bajo inspección.

Los diagramas de actividades son fuente de información poderosa para toda la gente que pueda entender el significado de los símbolos. Es importante adoptarlo cuando la gente está calificada en el uso de tales diagramas, como siempre, para usarlos solamente cuando estos son estrictamente necesarios.

Figura 10. Ejemplo de diagramas de actividad con bifurcación y unión para mostrar la ejecución paralela del flujo.

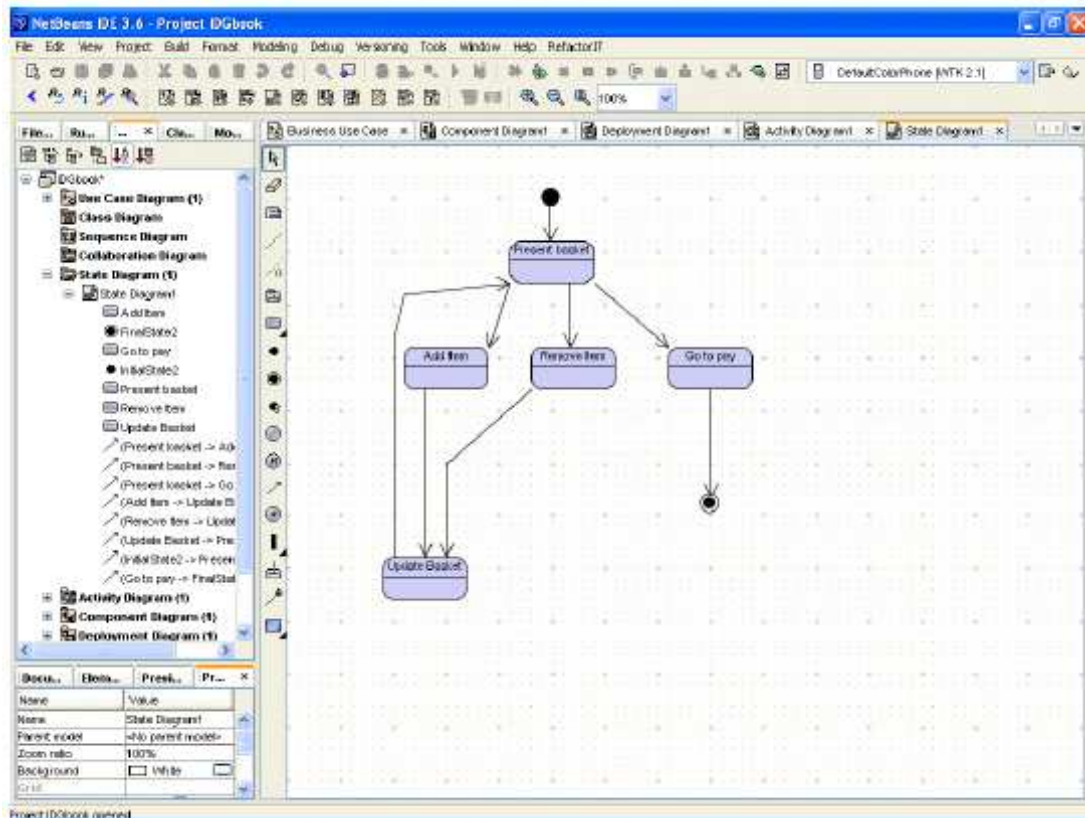


2.6.2.3 Adopción de diagramas de Estado

El diagrama de estados puede ser empleado útilmente para detallar el comportamiento de una clase que es particularmente compleja y dificulta el entendimiento o el conocimiento de la cual ha de ser compartida entre diferentes personas en el equipo. Un diagrama de estado inicia con los diferentes estados que la clase puede asumir durante su ciclo de vida y continúa definiendo la transición entre estados y las condiciones bajo las cuales es posible pasar desde un estado a otro.

Los estados que tienen que ser considerados son estados de alto nivel que son visibles desde el exterior de las clases. La implementación interna en los diagramas de estado usualmente no son mostrados, aunque nadie niega la posibilidad para describir la clase en un nivel más detallado.

Figura 11. Ejemplo de un diagrama de estado para la actualización de la canasta del cliente.



Como siempre, se tiene que recordar que se está desarrollando de acuerdo a los principios del desarrollo Agile, y por lo tanto, se tiene que documentar en el interior del sistema el código y el usar otras metodologías documentadas cuando el código por si mismo o la documentación automatizada no es suficiente para describir los procesos con lo deseado y claridad necesaria.

CAPITULO 3

HERRAMIENTAS

3.1 INTRODUCCION A LINUX

3.1.1 UN POCO DE HISTORIA

El nacimiento del sistema operativo Linux no ha sido fruto de la casualidad, sino todo lo contrario. Es el resultado de varios acontecimientos que se han sucedido en diferentes momentos a lo largo de las últimas décadas que podríamos resumir, principalmente, en los tres siguientes:

El primero de ellos se relaciona con la aparición del sistema operativo UNIX cuya gestión se inicia con los trabajos de Dennis Ritchie, durante los años 70, en los laboratorios de AT & T (American Telephone and Telegraph Corporation). En un principio estaba escrito en lenguaje ensamblador, aceptaba tan solo dos usuarios y recibió el nombre de UNICS. En 1973 se reescribió todo el código en lenguaje C, se amplió el número de usuarios y se le bautizó con el nombre de UNIX. Se distribuyó por universidades de todo el mundo. Una de éstas llegó a la Universidad de California en Berkeley la cual participó con muchas innovaciones a través de la BSD (Berkeley Software Foundation). Entre todos, se había construido un sistema operativo robusto y estable caracterizado por realizar los trabajos desglosándolos en múltiples y simple tareas que se ejecutan por separado pero de un modo seguro. En el año 1982 salieron al mercado las diferentes versiones: AIX de IBM, XENIX de Microsoft, UNIX BSD, etc. Unos años más tarde se homologaron todas las distribuciones bajo el mismo estándar UNIX SYSTEM V Versión 4. Su interfaz, por entonces, era solo alfanumérica (solo en modo texto). Por entonces, UNIX era un sistema que necesitaba de unos recursos de hardware muy potentes que estaban sólo al alcance de organizaciones militares, administrativas o académicas.

El segundo acontecimiento tuvo como punto de partida la FSF (Free Software Foundation) que con carácter no lucrativo nació en 1984. Su objetivo principal era crear un sistema operativo GNU, que se llamaría UNIX y que sería de libre

distribución. Otro éxito de la FSF fue el asentamiento de las bases de un nuevo tipo de licencia para el software. Es la llamada GPL (General Public License), que permite distribuir los programas de modo gratuito siempre que éstos se acompañen con el código fuente correspondiente. Hoy en día los términos GNU y GPL son prácticamente equivalentes.

Un tercer paso decisivo se produce en 1987 a raíz de la necesidad que el profesor de sistemas operativos Andrews S. Tanenbaum tenía para explicar a sus alumnos cómo funciona por dentro un sistema operativo. Al no disponer de suficiente información sobre los sistemas de software propietarios que había, por aquellos años, optó por escribir un sistema operativo muy sencillo publicando, al mismo tiempo, todo el código fuente. Le llamó MINIX por su parecido con UNIX y su sistema de archivos "minix" todavía se emplea hoy en día debido a su elevada eficacia, sobre todo, en dispositivos de poca capacidad como disquetes o discos-ram. La idea de Tanenbaum le gustó mucho a un estudiante finlandés de informática llamado Linus Torvalds quien tenía en mente crear un sistema operativo como UNIX pero que fuese capaz de adaptarse al hardware de un ordenador personal. Linus, además, tuvo otra buena idea, usar la incipiente Internet para dar a conocer su proyecto, bajo licencia GPL y a todo el mundo, el 5 de octubre de 1991 desde la Universidad de Helsinki, comienza así la andadura y el desarrollo de un sistema operativo edificado, desde el primer momento, sobre las necesidades, la creatividad y la participación de sus mismos usuarios.

Desde entonces, el crecimiento, uso y aumento de prestaciones de Linux no se ha detenido gracias al elevado número de desarrolladores, colaboradores altruistas y usuarios de todo el mundo. Se utiliza en empresas, administraciones y usuarios domésticos, ofreciendo una alternativa al software comercial de la competencia. Sin embargo, donde realmente brilla por sus cualidades es en el sector educativo. Hay multitud de información al respecto. Solo la editorial Prentice Hall tiene más de 100 títulos publicados. Existen en nuestro país varias revistas dedicadas solo a Linux y otras dedican varias páginas a temas relacionados directamente con él. En la red existen libros enteros gratuitos, así como gran cantidad de documentación que permite hacer cualquier cosa a cualquier usuario que se lo

proponga. Se puede afirmar que, en Linux, no hay nada oculto y que toda la información esta a disposición de quien la necesite.

3.1.2 ¿QUE ES LINUX?

En principio se puede asumir que este nombre se reserva para nombrar al núcleo, o kernel, del sistema operativo en sí y poco más. El núcleo aparece en la mayor parte de los sistemas como un archivo de nombre vmlinuz que se carga en la memoria RAM del ordenador durante el proceso de arranque bien sea a través de disquete, de CDROM o de disco duro.

El kernel es un programa que "envuelve" al conjunto de los elementos físicos, o hardware, que componen el ordenador, haciendo de intermediario, para que el usuario no tenga que preocuparse demasiado por el funcionamiento de éstos. Se encarga también de planificar la ejecución de los procesos o el uso de los recursos del sistema, supervisar la transmisión de los datos entre aplicaciones y los dispositivos periféricos. Una parte importante del núcleo lo constituyen los controladores de dispositivos, o drivers, que pueden incluirse formando parte del mismo durante el proceso de compilación o bien pueden acompañarlo por separado, como archivos binarios de código objeto, que se cargarán después en memoria sólo aquellos que sean necesarios para controlar un dispositivos físico concreto, lo cual hace que el núcleo sea más pequeño y ligero. Como es de esperar, los controladores, deben compilarse al mismo tiempo que el núcleo sobre el que luego funcionarán, caso de acompañarlo como módulos independientes.

Otra particularidad del kernel es la posibilidad de que cualquier usuario pueda utilizar el código fuente escrito en lenguaje C y compilarlo en su mismo ordenador introduciendo las modificaciones que estime necesarias para un uso personalizado. En efecto, durante el proceso de compilación, se pueden seleccionar gran número de opciones, de un modo muy sencillo, a través de un menú gráfico, o en modo texto, y sin necesidad de tener conocimientos avanzados de programación.

Aunque parezca extraño, el kernel sólo, no sirve para nada. El primer elemento que resulta imprescindible es el intérprete de comandos, que es otro archivo independiente del núcleo. Recibe el nombre de shell porque actúa como una coraza que envuelve al núcleo. Es una interfaz que permite al usuario comunicarse con el sistema operativo. Al igual que en UNIX, en Linux, hay disponibles varios shell's. El más usual es el bash que es GNU (de la FSF). Utilizando el shell pueden realizarse programas bastante completos que se llaman guiones (scripts). Las órdenes que acepta el shell, a través de la línea de comandos, están básicamente formadas por el nombre de la orden, uno o varios modificadores precedidos por un guión y uno o varios argumentos, por ejemplo, nombres de archivos o directorios. Estos campos deberán ir siempre separados por un espacio.

\$ orden -modificador argumento Ejemplo: \$ ls -l /tmp

Resumiendo las principales especificaciones técnicas de Linux, como sistema operativo, se puede decir que es:

Multiusuario. Varios usuarios, bajo la supervisión de un sólo S O, comparten al mismo tiempo todos los recursos del ordenador, microprocesador, RAM, discos, impresora, etc (siempre que cada uno tenga su propio terminal formado por pantalla y teclado).

Multitarea real. Cada uno de los posibles usuarios conectados al sistema, puede ejecutar, a su vez, varios trabajos al mismo tiempo. A mayor número de procesos iniciados, más tardarán en completarse, pero se terminarán bien.

Multiplataforma. Al estar escrito en lenguaje C, se puede compilar para diferentes arquitecturas, por ejemplo: Intel, Motorola, Alpha, Sparc, etc.

Multiprocesador. Puede funcionar en ordenadores que tengan una placa base con más de un procesador. Admite arquitecturas de 32 y 64 bits.

Servidor de red. Puede gestionar el acceso y el uso de los recursos compartidos de una red local compuesta por otras máquinas sean Linux o no. Dispone de los servicios necesarios para trabajar con redes externas, como router, prestar servicios de seguridad, o firewall, servidor ftp, etc. Es un sistema orientado hacia las redes.

Interfaz alfanumérica. Sin duda alguna es el sistema operativo más potente en lo que a interfaz en línea de comandos se refiere. Además, puede utilizar interfaz gráfica, llamada X-Windows, sobre la cual podemos instalar dos tipos de escritorios diferentes a elegir: el KDE o el GNOME, ambos tan funcionales e intuitivos como los de los sistemas operativos de la competencia, pero GNU.

Estabilidad y seguridad. Una vez instalado, y correctamente configurado, es el sistema más estable y seguro, característica que comparte junto con su antecesor UNIX.

Gratis. En general, no es necesario realizar ninguna inversión para adquirir tanto el SO como la mayor parte de aplicaciones para Linux que también son gratuitas. Analizaremos con mayor detalle estos aspectos en el apartado siguiente.

3.1.3 REQUISITOS DE LA PLACA BASE Y DEL PROCESADOR

Actualmente, Linux soporta sistemas con una CPU Intel 80386, 80486, o Pentium, y un número, cada vez mayor, de procesadores.

Los «clónicos» no Intel, como AMD, también funcionan con Linux. Si tiene la suerte de tener una placa para doble microprocesador, Linux la detectará y hará funcionar los dos «micros» sin ningún problema.

Si se conserva un antiguo 386 ó 486, puede que lo quiera aprovechar con Linux. No va a alcanzar velocidades de vértigo, pero tendrá una pequeña estación de trabajo conectada en red, incluso ejecutar aplicaciones usando los recursos del PIII. El coprocesador matemático puede ser emulado por el kernel si la máquina no cuenta con uno.

3.1.4 REQUISITOS DE LA MEMORIA RAM

Linux utilizará toda la memoria RAM de forma automática, y se acoplará sin problemas con tan sólo 128 Mb de RAM, incluyendo un entorno de ventanas sencillo.

En las configuraciones actuales es fácil contar con 256 Mb, es suficiente para un uso personal; para servidores dependerá de la carga que esperemos. Disponer de más memoria es tan importante como tener un procesador más rápido.

Linux reserva una parte del disco duro para espacio de intercambio (*swap*) que se usa como RAM virtual, incluso si dispone de bastante memoria RAM física en su máquina. El área de *swap* no puede compararse en términos de velocidad a una memoria RAM, pero permite a su sistema ejecutar aplicaciones más grandes guardando en disco duro partes de código que están inactivas. La cantidad de espacio de intercambio a reservar depende de diversos factores; se puede hacer un cálculo en función de la RAM física, pero depende del sistema.

Normalmente 16 Mb de *swap* deberían ser suficientes y 64 Mb estarán francamente bien. Lo que no se aconseja que haga nunca es dejar al sistema sin espacio de intercambio.

3.1.5 REQUISITOS DE ESPACIO EN EL DISCO DURO

La cantidad de espacio depende en gran medida de sus necesidades y de la cantidad de software que quiera instalar. Las distribuciones Linux incluyen gran cantidad de software que quizá no necesite utilizar. También puede disponer de espacio para Linux en múltiples unidades.

Un sistema básico funcionará con menos de 1 Gigabyte.

Una instalación media, con entorno gráfico y un número de aplicaciones respetable, van bien con 2 Gb y da cabida a expansiones y espacio para que los usuarios dejen ficheros.

Una distribución completa ocupa alrededor de 5 Gb.

Con el tamaño actual de discos, estas cantidades no deberían suponer gran problema.

3.1.6 CONTROLADOR DE DISPOSITIVOS

El núcleo soporta controladoras IDE, SATA y SCSI. La regla general para controladoras que no sean SCSI, es que si puede acceder a las unidades (disco y/o disquete) desde MS-DOS u otro sistema operativo, debería poder hacerlo desde Linux.

3.1.7 BUSES Y TIPOS DE PUERTOS SOPORTADOS POR LINUX

Todos los buses y puertos que existen en la actualidad están soportados por linux incluso los últimos como el USB, Firewire.

3.1.8 REQUISITOS DEL SISTEMA GRAFICO

Para trabajar en modo texto, Linux soporta todas las tarjetas de vídeo estándar VGA, IBM monocromo y Super VGA. Los entornos gráficos como el Sistema X Window tienen requisitos propios de hardware para la tarjeta de vídeo. Se soportan prácticamente todas las tarjetas actuales, varias aceleradoras 3D.

3.1.9 SOPORTE DE DISPOSITIVOS DE CONEXIÓN A LA RED LOCAL

Linux soporta un buen número de tarjetas Ethernet y adaptadores para LAN, además dispositivos Wireless. La lista es demasiado.

3.1.10 SOPORTE DE IMPRESORAS

Linux soporta prácticamente todas las impresoras paralelas. El software de impresión de Linux se basa en el estándar de UNIX lp y lpr. Este software también le permite imprimir remotamente a través de la red, si es que tiene una disponible.

3.1.11 ESTRUCTURA

La estructura de Linux semeja un árbol donde cada partición y dispositivo de lectura/escritura se representa como un directorio, los nombres de las unidades de disco duro son:

hda: disco duro principal

hdb: disco duro secundario

hda1: primera partición del disco duro principal.

hdb2: segunda partición del disco secundario.

3.2. ACERCA DEL PROTOCOLO HTTP.

HTTP (Hypertext Transfer Protocol, o Protocolo de Tránsito de Hipertext), es el método utilizado para transferir o transportar información en la Red Mundial (WWW, World Wide Web). Su propósito original fue el proveer una forma de publicar y recuperar documentos HTML.

El desarrollo del protocolo fue coordinado por World Wide Web Consortium y la IETF (Internet Engineering Task Force, o Fuerza de Trabajo en Ingeniería de Internet), culminando con la publicación de varios RFC (Request For Comments), de entre los que destaca el RFC 2616.

HTTP es un protocolo de solicitud y respuesta a través de TCP, entre agentes de usuario (Navegadores, motor de índice y otras herramientas) y servidores, regularmente utilizando el puerto 80. Entre la comunicación entre éstos puede intervenir como servidores Intermediarios (Proxies), puertas de enlace y túneles.

3.3 APACHE.

3.3.1 ¿Qué es Apache?

Apache es un servidor HTTP, de código abierto y licenciamiento libre, que funciona en Linux, sistemas operativos derivados de Unix™, Windows, Novell Netware y otras plataformas. Ha desempeñado un papel muy importante en el crecimiento de la red mundial, y continua siendo el servidor HTTP más utilizado, siendo además el servidor *de facto* contra el cual se realizan las pruebas comparativas y de desempeño para otros productos competidores. Apache es desarrollado y mantenido por una comunidad de desarrolladores auspiciada por Apache Software Foundation.

URL: <http://www.apache.org/>

3.3.2 ¿Cómo se comporta Apache en relación a las pruebas comparativas con otros servidores de la web?

Las pruebas dependen mucho de cómo se efectúan y sobre todo de por quién. Las estadísticas más fiables parecen favorecer a Apache.

Apache se demuestra más rápido que otros Web Server free; algunos Web Server comerciales, sin embargo, parece que son más veloces que Apache, pero aquí entra en juego el discurso anterior a propósito de "Quién hace los benchmark".

En cualquier caso, no es una casualidad que Apache sea el Web Server más difundido en la gran red (se habla, aproximadamente, de 1.500.000 server) se puede encontrar estas estadísticas en la red.

3.3.3 ¿Por qué precisamente Apache con todos los servidores de Web que se encuentran en circulación?

Sobre todo por cuestión de gustos, pero no sólo por eso. Muchos de los servidores de Web que se pueden encontrar son comerciales y, por eso mismo, más adecuados para usuarios profesionales que pueden permitirse inversiones en este sentido; además, un software profesional no es a priori mejor que uno libre: es más, a veces sucede justo al contrario.

Otra razón para preferir Apache es la gran difusión en los servidores Web, que garantiza rendimiento y estabilidad.

Y además Apache es un proyecto, y como proyecto, está abierto a las críticas, parches y bugfixes sugeridos directamente por sus usuarios, que pueden entrar a formar parte del equipo de desarrollo; asimismo, la posibilidad de contar con un software con tantas fuentes es algo digno de tenerse en cuenta. En último término, depende del sistema en el que se le debe hacer trabajar: en los Unix, Apache es probablemente la mejor elección; en los sistemas Windows, sin embargo, parece más conveniente, al menos por el momento, dirigirse a IIS, dada la completa integración con el sistema (se habla siempre de instrumentos firmados por Microsoft).

3.3.4 ¿Qué hacer para configurar Apache?

Apache se basa fundamentalmente en dos archivos, `httpd.conf` y `srm.conf`, presentes en la `RootDir` (ver más) del servidor. Después, para una configuración más precisa, se pueden crear ad hoc archivos que determinen directorios, de manera que Apache actúe en consecuencia una vez que lea las informaciones que encuentre en ellos.

De este modo, será posible darle al servidor de Web una configuración base y un directorio personalizado.

3.4 MySQL

3.4.1 ¿Qué es MySQL?

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

3.4.2 Historia de MySQL

MySQL surgió como un intento de conectar el gestor `mSQL` a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que `mSQL` no era lo bastante flexible para lo que necesitaban, por

lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

No se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

3.4.3 Características de MySQL

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

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

3.4.4 Lo mejor de MySQL

Es evidente que la gran mayoría de gente usa este gestor en Internet, por lo que encontrar opiniones favorables no ha resultado en absoluto complicado:

Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.

Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.

Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.

Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.

El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet y de buscadores de aplicaciones.

3.5 PHP

3.5.1 Historia de PHP

PHP comenzó y sigue siendo primeramente usado como un lenguaje de script del lado del servidor embebido en HTML.

PHP, se conoce originalmente como Personal Home Pages, fue concebido en el otoño de 1994 por Rasmus Lerdorf. El lo escribió como una forma de track visitantes a su CV en línea. La primera versión salió en los comienzos de 1995, y fue ahí donde Rasmus se dio cuenta que haciendo en proyecto código-abierto, las personas arreglarían sus problemas. La primera versión fue muy precaria y tenía un parser que reconocía solo unas pocas macros y brindaba algunas utilidades que se usaban comúnmente en sitios Web.

El parser fue reescrito a mediados de 1995 y se lo renombró a PHP/FI versión 2. El "FI" en esta versión quería decir Interpreter Formal. Lo que Rasmus había agregado a PHP fue de acuerdo a las necesidades crecientes de sitios Web. El soporte para mSQL fue agregado. PHP/FI tuvo un crecimiento masivo, y otra gente empezó a contribuir programando regularmente.

A mediados de 1997 Zeev Suraski y Andi Gutmans reescribieron el parser principal, y PHP cambió de estar en manos de Rasmus a un grupo más orientado al proyecto. Esto formó las bases para que PHP3, fuere ahora llamado PHP: Hypertext Preprocessor - un acrónimo recursivo.

La versión de PHP4, es otra reescritura de Suraski and Gutmans y esta basada en el motor Zend. PHP ahora tiene doscientos contribuyentes regularmente trabajando en varias partes del proyecto. Tiene una cantidad muy grande extensiones, módulos y soporta todos los servidores más populares nativamente, y además tiene soporte para MySql y ODBC.

Las últimas estadísticas muestran que PHP es actualmente usado por más de 5.5 millones de dominios, y ha tenido un gran crecimiento durante el último año. Es lejos el módulo más popular de Apache; para dar alguna perspectiva, Apache actualmente tiene un 60% del mercado de servidores de Internet, y el servidor IIS (con soporte nativo para ASP) tiene menos de la mitad de esa proporción del mercado.

3.5.2 ¿Qué hace PHP?

PHP es un lenguaje de propósito general. Es normalmente usado como un lenguaje de script embebido en HTML para su uso en la Web, pero puede también puede usarse como un lenguaje de script para shell o hasta como un lenguaje para escribir aplicaciones con ventanas, con PHP-GTK.

Debido a la naturaleza open-source de PHP, si hay algo que actualmente no se pueda hacer en PHP no hay ningún impedimento para que vos puedas escribir un módulo o una extensión en código C para extender la funcionalidad y puedas utilizar PHP con lo que vos quieras. Esto es posible por la buena documentación de la API que esta disponible para todos.

PHP4 tiene dos partes:

El motor Zend

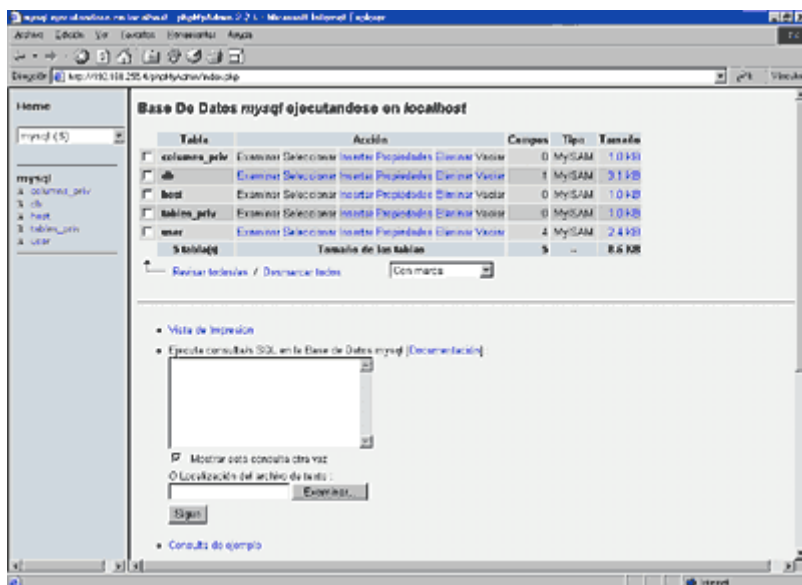
es la parte del paquete PHP que mantiene los pedidos, los procesos de los archivos de script y maneja las variable y los recursos. Su sitio puede encontrarse en <http://www.zend.com/>.

PHP

implementa el 90% de la funcionalidad que ve el usuario final. Brinda un amplio rango de módulos como el soporte para MySQL, ODBC and XML

3.6 PHPMYADMIN

Un proyecto de código abierto en PHP para administrar la base de datos MySQL a través de una interfaz Web. PhpMyAdmin es un programa de libre distribución escrito en PHP. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz Web muy intuitiva.



La aplicación en si no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor Web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas. La herramienta nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

Página de phpMyAdmin La página de inicio del proyecto es <http://www.phpmyadmin.net/>. Desde allí podemos descargar los ficheros de la última versión de la aplicación, que posteriormente debemos colocar en nuestro servidor web. También podemos encontrar a phpMyAdmin dentro de la red Sourceforge.net, que es un sitio que recoge multitud de proyectos "Open Source" (código abierto).

Hay varias versiones disponibles, pero es recomendable escoger la que nos aconsejen como la última versión estable (The last stable versión). En el momento de escribir esta tesis se usa la versión 2.11.5.1.

Los archivos que hemos descargado son de la versión 4 de PHP, aunque también ofrecen la posibilidad de bajarse los archivos que guardan compatibilidad con la versión 3 de PHP, para que aquellos que no dispongan del motor de PHP más actual.

Home

desarrolloweb (9)

desarrolloweb

- academia_seccion
- seccion_curso
- tacademia
- tcertificados
- tcursos
- tmetodologia
- tprovincias
- tseccion
- ttipos

Base De Datos *desarrolloweb* - tabla *tacademia* ejec

Mostrando campos 0 - 2 (3 total)

SQL-query : [Editar]
SELECT * FROM `tacademia` LIMIT 0, 30

Mostrar: 30 filas empezando de 0
en modo horizontal y repite encabezados cada 100 celdas

		id_academia	nombre	contrasena	persona	en
Editar	Borrar	1	academia12	password12	persona12	em
Editar	Borrar	2	academia2	password2	persona2	em
Editar	Borrar	9	333	333	33	33

Mostrar: 30 filas empezando de 0
en modo horizontal y repite encabezados cada 100 celdas

3.7 FLASH

El software Adobe Flash 8 es el entorno de creación más avanzado para generar contenido interactivo de alta calidad para plataformas móviles, digitales y Web. Crea sitios Web interactivos, anuncios con abundante contenido multimedia, medios didácticos, atractivas presentaciones, juegos y mucho más. Los desarrolladores y diseñadores que trabajan con los sistemas Macintosh y Windows®.

Flash CS3 Professional es una aplicación disponible para Microsoft® Windows®, y para Mac como aplicación binaria universal, que proporciona las herramientas necesarias para llevar a cabo proyectos con éxito.

3.7.1 Línea de tiempo por fotogramas

Añade movimiento a las creaciones de forma rápida con la línea de tiempo por fotogramas fácil de usar y altamente controlable, creada mediante principios de animación tradicionales tales como fotogramas clave y películas.

3.7.2 Dibujos y formas

Crea fácilmente formas, redondea las esquinas de rectángulos, determina el radio de un círculo y mucho más. Ajusta visualmente las propiedades de forma en cada fase.

3.7.3 Desarrollo de ActionScript 2.0

Ahorra tiempo con el lenguaje ActionScript 2.0, y proporciona un rendimiento mejorado, mayor flexibilidad y un desarrollo más intuitivo y estructurado.

3.7.4 Sofisticadas herramientas de vídeo

Crea, edita y desarrolla contenido de FLV de transferencia y descarga progresiva con compatibilidad completa con vídeo. Asegura la mejor experiencia de vídeo con un codificador de vídeo independiente, soporte para canales alfa, vídeo códec de alta calidad, puntos de inicio incorporados, compatibilidad de importación de vídeo.

3.7.5 Amplias funciones de diseño

Ajusta visualmente las propiedades de forma en cada fase mediante las herramientas prácticas de diseño de formas, crea exhaustivas ilustraciones de vectores con la nueva herramienta Pluma inspirada en Adobe Illustrator, pega ilustraciones desde Illustrator CS3 a Flash 8 y mucho más.

3.7.6 Arquitectura ampliable

Aprovecha las interfaces de programación de aplicaciones (API) de Flash para desarrollar fácilmente extensiones que incorporen funcionalidad personalizada.

3.8 DREAMWEAVER

3.8.1 Qué es Dreamweaver 8?

Diseña, desarrolla y realiza el mantenimiento integral de sitios y aplicaciones Web de forma rápida y sencilla con Adobe Dreamweaver 8. Creado para diseñadores y desarrolladores, Dreamweaver 8 ofrece la posibilidad de trabajar en una interfaz de diseño visual intuitiva o en un entorno de codificación agilizado. La integración inteligente con el software Adobe Flash 8 Professional le garantiza un flujo de trabajo eficaz.

3.8.2 Crea contenido web sofisticado

Está surgiendo un estándar superior en diseño y desarrollo web proporcionado por Dreamweaver 8. Crea interfaces dinámicas y sofisticadas que integren estrechamente estructuras estáticas e interactividad receptiva.

Acelera su flujo de trabajo con Dreamweaver 8, una versión más rápida y homogénea de la herramienta de diseño y desarrollo web preferida. Coloque archivos FLV. Crea de forma visual o genera el código directamente, trabaja en Macintosh o Windows.

3.8.3 Prácticas recomendadas

Dreamweaver 8 facilita la aplicación de prácticas recomendadas y estándares aceptados para garantizar que sus sitios web y aplicaciones se complementan entre sí. Crea sitios web basados en CSS desde la base con nuevos diseños de CSS, el panel de CSS unificado y las herramientas de visualización y gestión de CSS. Garantiza un rendimiento más fiable y coherente con la nueva función de comprobación de compatibilidad con el explorador.

3.9 RATIONAL SYSTEMS DEVELOPER

Rational Systems Developer es una herramienta de diseño y desarrollo que saca provecho de todas las posibilidades de **Eclipse** e incluye plug-ins que permiten el desarrollo dirigido por modelos y una arquitectura de software para la creación de aplicaciones con arquitecturas sólidas basadas en Corba, C/C++ y Java J2SE, mediante **Unified Modeling Language (UML)**).

- Ofrece generación de código para C++, Java, CORBA IDL, XSD
- Da soporte a la iniciativa MDA de OMG
- Da soporte al diseño y desarrollo de varios lenguajes en el mismo shell
- Proporciona funciones de análisis estructural para descubrir la arquitectura de una aplicación
- Da soporte al desarrollo dirigido por modelos mediante **UML 2.0**, al desarrollo de metamodelos personalizados y a la extensibilidad de herramientas para crear aplicaciones con arquitecturas modificables
- Se integra con otras funciones de Rational Software Development Platform.

3.10 CentOS 5.

3.10.1 CONSIDERACIONES PREVIAS A LA INSTALACION DE CENTOS 5

En realidad, la instalación de Linux no es un proceso complicado, y con la nueva instalación en modo gráfico, completar con éxito el proceso es relativamente fácil, incluso con pocos conocimientos del sistema. Si tiene otro sistema operativo instalado en el disco duro, necesitará reparticionarlo, a continuación veremos un par de puntos importantes sobre las necesidades de particionamiento de Linux, el potente gestor de particiones fdisk y el posterior uso de las particiones.

3.10.2 DISPOSITIVOS Y PARTICIONES EN LINUX

Los discos duros se encuentran divididos en particiones, donde cada partición corresponde a un sistema operativo.

En el primer sector del disco está el registro de arranque maestro junto a la tabla de particiones. El registro de arranque (como su nombre indica) se usa para arrancar el sistema. La tabla de particiones contiene información acerca del lugar y el tamaño de cada partición. Hay tres clases de particiones:

Primarias: Las más usadas, debido al límite del tamaño de la tabla de particiones, sólo pueden tenerse hasta cuatro particiones primarias en un disco.

Extendidas: Una partición extendida no tiene datos ella misma; en su lugar, actúa como "soporte" de particiones lógicas; solo puede tenerse una partición extendida por disco.

Lógicas: Se puede crear cualquier número de particiones lógicas.

Linux requiere por lo menos una partición, para el sistema de archivos raíz. Si desea crear varios sistemas de archivos, necesitará una partición por cada sistema de archivos.

La otra cuestión a considerar es el espacio de intercambio (swap). Se debe crear una partición de intercambio, una partición reservada exclusivamente como espacio de intercambio.

Por lo general, se crearán, al menos, dos particiones para Linux: una para ser usada como sistema de archivos raíz, y la otra como espacio de intercambio. Por supuesto, hay otras opciones, pero ésta es la opción mínima.

El tamaño de los sistemas de archivos de su sistema Linux depende en gran parte de qué software quiera instalar en él. El tamaño de sus particiones de intercambio (debe elegir una para esto) depende de la RAM virtual que necesite.

Muchas distribuciones necesitan que se creen a mano las particiones de Linux utilizando el programa fdisk. Otras pueden crearlas durante el mismo proceso de instalación. En cualquier caso, debemos conocer lo siguiente acerca de los nombres para los dispositivos y las particiones en Linux. Bajo Linux, los dispositivos y las particiones tienen nombres muy distintos a los utilizados en otros sistemas operativos.

En MS-DOS / Windows, las disqueteras se identifican como A: y B:, mientras que las particiones del disco duro se identifican como C:, D:, etc. Con Linux, la denominación es algo diferente.

Dispositivo	en Linux
Primera disquetera (A:)	/dev/fd0
Primer disco duro (todo el disco)	/dev/hda
Primer disco duro, partición primaria 1	/dev/hda1
Primer disco duro, partición primaria 2	/dev/hda2
Primer disco duro, partición lógica 1	/dev/hda5
Primer disco duro, partición lógica 2	/dev/hda6
Segundo disco duro (todo el disco)	/dev/hdb
Segundo disco duro, partición primaria 1	/dev/hdb1
Primer disco duro SCSI (todo el disco)	/dev/sda
Primer disco duro SCSI, partición primaria 1	/dev/sda1
Segundo disco duro SCSI (todo el disco)	/dev/sdb
Segundo disco duro SCSI, partición primaria 1	/dev/sdb1

Obsérvese que /dev/fd0 corresponde a la primera disquetera (A: bajo Windows) y que /dev/fd1 es la segunda (B:). Los discos duros SCSI se nombran de manera diferente a otros discos. Los IDE y CD-ROM (que no sean SCSI) se acceden a través de /dev/hda, /dev/hdb, etc. Las particiones de /dev/hda son /dev/hda1, /dev/hda2, etc. Los dispositivos SCSI son /dev/sda, /dev/sdb, etc., y las particiones con /dev/sda1, /dev/sda2, etc. Las particiones lógicas se nombran de forma consecutiva partiendo de /dev/hda5.

Arranque de CD-ROM: Primeramente debe asegurarse que su BIOS se encuentre configurado para leer el CD-ROM/ DVD ROM al momento de inicializarse su PC o Servidor, esto es necesario para que se ejecute el "shell" de instalación CentOS en lugar del sistema operativo que tiene.

3.10.3 CARACTERISTICAS DEL HARDWARE USADO PARA LA INSTALACION:

Si bien los sistemas GNU/Linux pueden instalarse en equipo con capacidades muy reducidas (o limitadas), para tener un entorno con un buen desempeño y que soporte las nuevas características de los sistemas incluidos en esta distribución, nuestro equipo tiene las siguientes características:

Procesador Intel Pentium4 1.6 GHZ

1 GB RAM

20 GB en disco duro IDE

Interfaz de red 10/100.

Tarjeta de Video compatible

3.10.4 INSTALACION

Inserte el disco DVD de instalación de CentOS 5 y en cuanto aparezca el diálogo de inicio (boot:), pulse la tecla ENTER o bien ingrese las opciones de instalación deseadas.



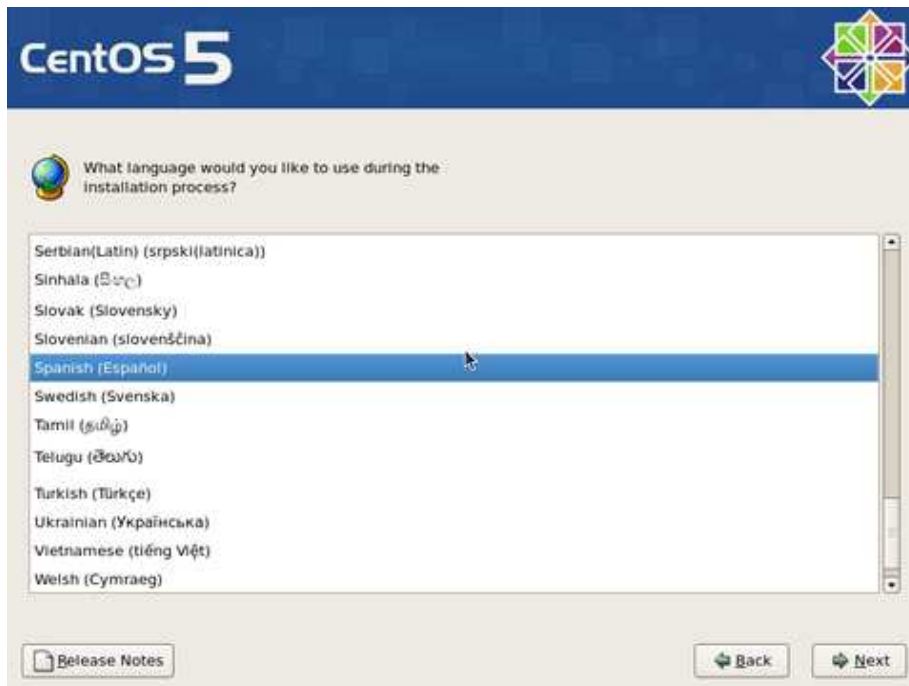
Si desea verificar la integridad del disco a partir del cual se realizará la instalación, seleccione «OK» y pulse la tecla ENTER, considere que esto puede demorar varios minutos. Si está seguro de que el disco o discos a partir de los cuales se realizará la instalación están en buen estado, seleccione «Skip» y pulse la tecla ENTER.



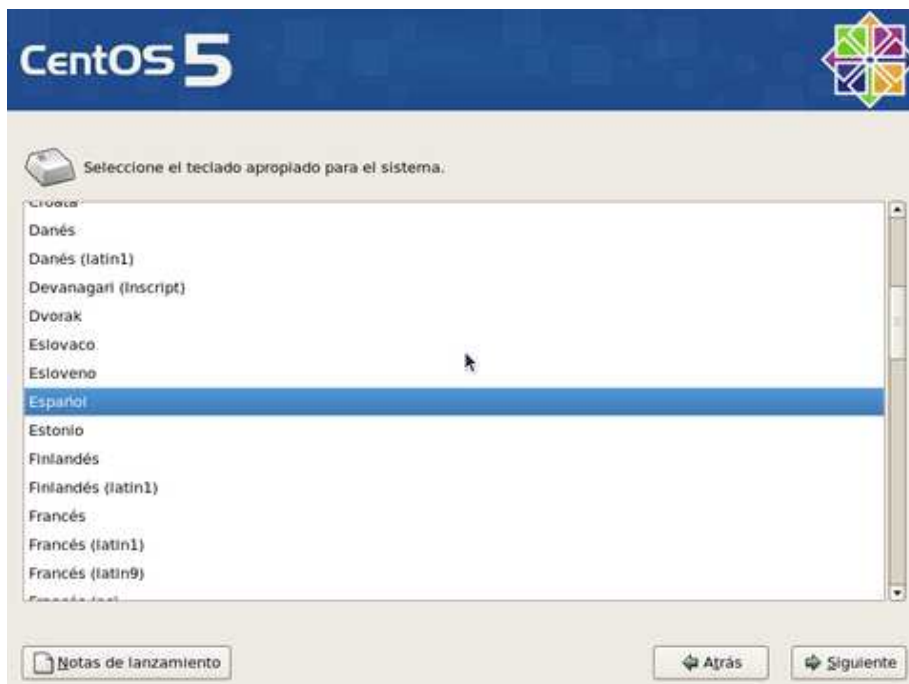
Haga clic sobre el botón «Next» en cuanto aparezca la pantalla de bienvenida de CentOS.



Seleccione «Spanish» como idioma para ser utilizado durante la instalación.



Seleccione el mapa de teclado que corresponda al dispositivo utilizado. El mapa «Español» o bien «Latinoamericano» de acuerdo a lo que corresponda. Al terminar, haga clic sobre el botón «Siguiente».



Salvo que exista una instalación previa que se desee actualizar (no recomendado), deje seleccionado «Instalar CentOS» y haga clic en el botón «Siguiente» a fin de realizar una instalación nueva.



Para crear las particiones de forma automática, lo cual puede funcionar para la mayoría de los usuarios, puede seleccionar:

- «Remover particiones en dispositivos seleccionados y crear disposición», lo cual eliminaría cualquier partición de cualquier otro sistema operativo presente, y creará de forma automática las particiones necesarias.
- «Remover particiones de linux en dispositivos seleccionados y crear disposición», lo cual eliminaría cualquier partición otra instalación de Linux presente, y creará de forma automática las particiones necesarias.
- «Usar espacio disponible en dispositivos seleccionados y crear disposición», lo cual creará de forma automática las particiones necesarias en el espacio disponible.



La instalación requiere la partición de su disco duro. Por defecto, una capa de partición razonable es escogida, ésta es suficiente en la mayoría de los casos. Usted puede escoger esta partición predeterminada o crearla usted mismo.

Crear disposición personalizada.

Seleccione la(s) unidad(es) a usar para esta instalación.

hda 8189 MB VMware Virtual IDE Hard Drive

[+ Configuración Avanzada de almacenamiento](#)

Revise y modifique la capa de particiones

[Notas de lanzamiento](#)

[← Atrás](#)

[→ Siguiente](#)

Conviene crear una disposición que permita un mayor control. Seleccione «Crear disposición personalizada».



La instalación requiere la partición de su disco duro. Por defecto, una capa de partición razonable es escogida, ésta es suficiente en la mayoría de los casos. Usted puede escoger esta partición predeterminada o crearla usted mismo.

- Remover particiones en dispositivos seleccionados y crear disposición.
- Remover particiones de linux en dispositivos seleccionados y crear disposición.
- Usar espacio disponible en dispositivos seleccionados y crear disposición.
- Crear disposición personalizada.**

hda 8189 MB VMware Virtual IDE Hard Drive

[+ Configuración Avanzada de almacenamiento](#)

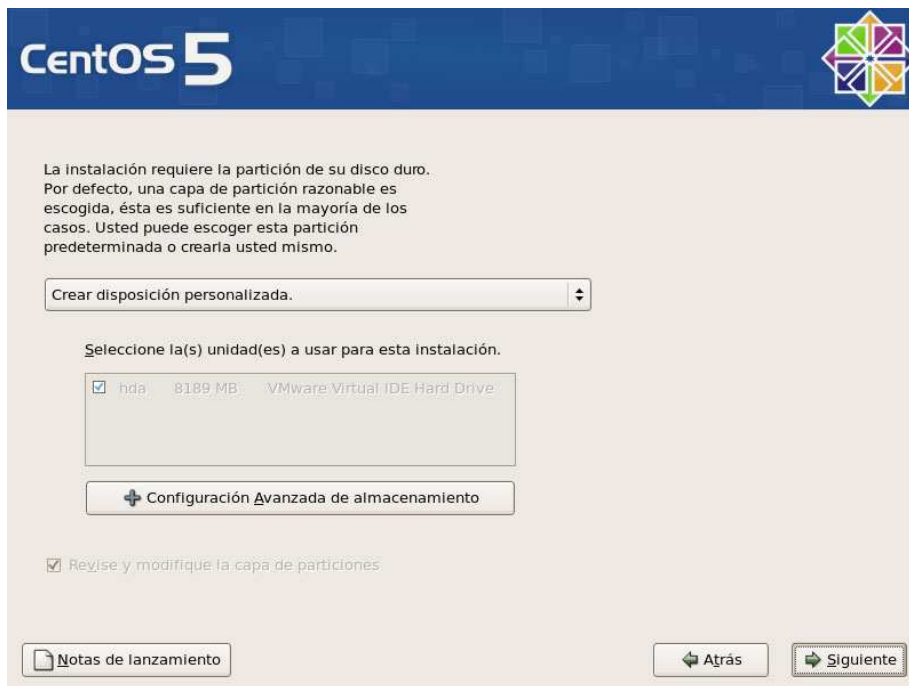
Revise y modifique la capa de particiones

[Notas de lanzamiento](#)

[← Atrás](#)

[→ Siguiente](#)

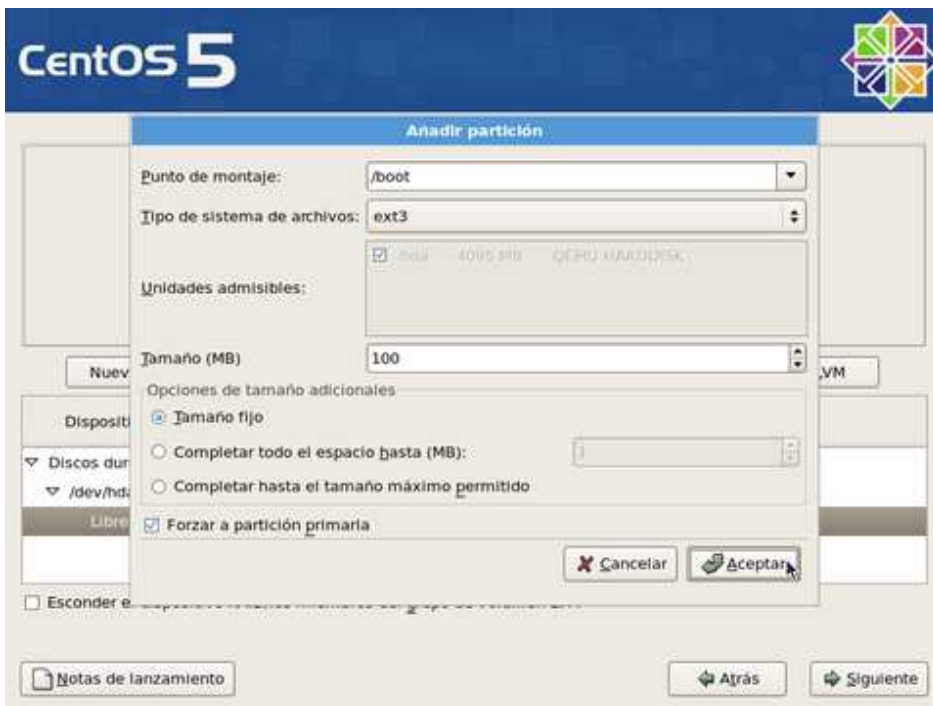
Una vez seleccionado «Crear disposición personalizada», haga clic sobre el botón «Siguiente».



La herramienta de particiones mostrará el espacio disponible. Haga clic en el botón «Nuevo».



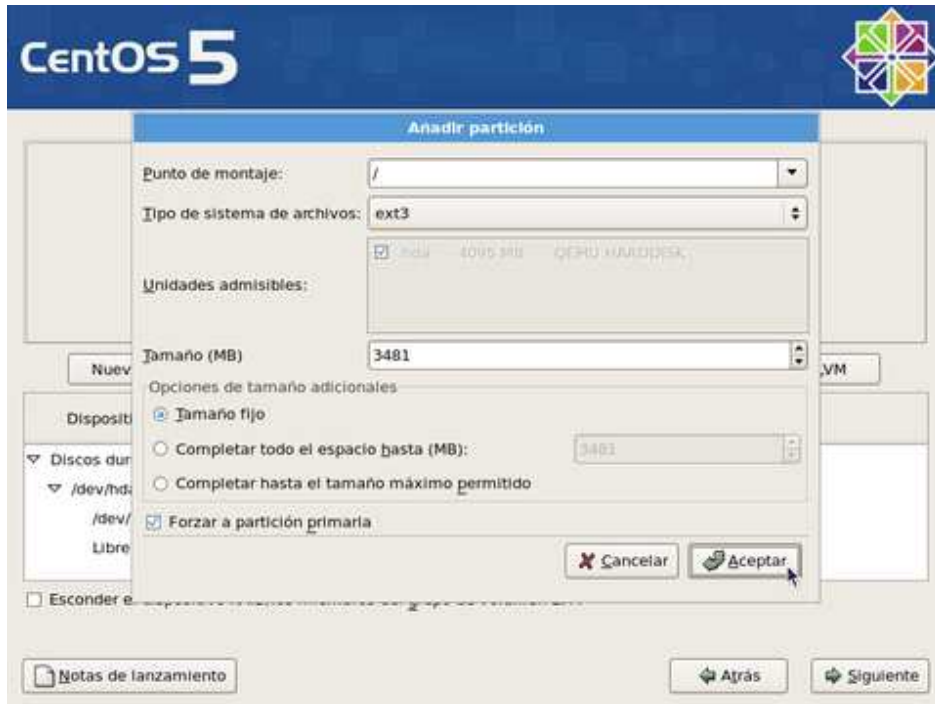
Asigne 100 MB a la partición /boot y defina ésta como partición primaria, siempre que la tabla de particiones lo permita.



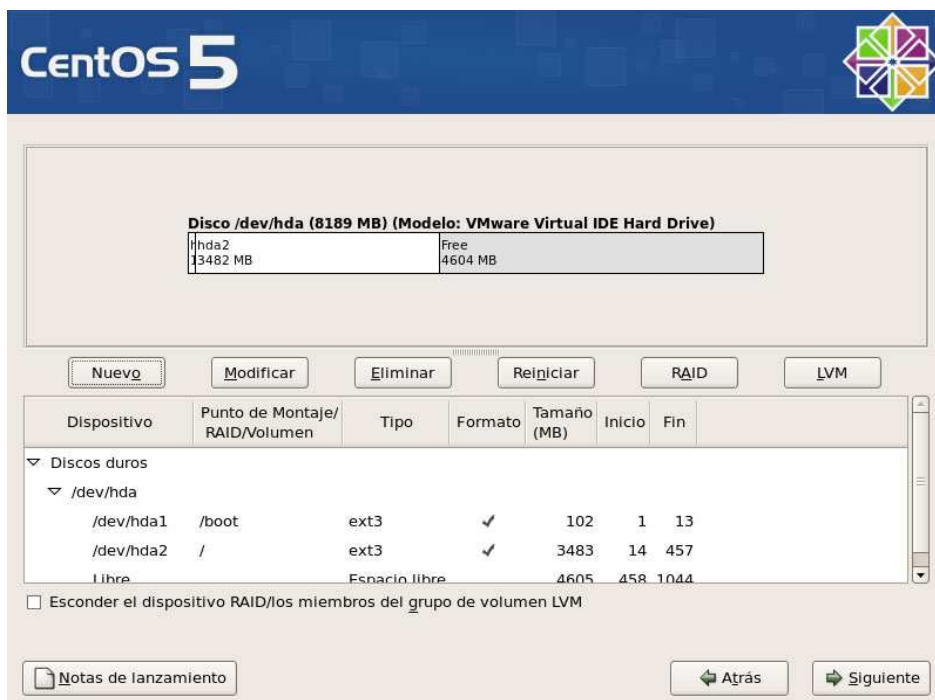
Si está conforme, haga clic otra vez en el botón «Nuevo» y proceda a crear la siguiente partición.



Asigne a la partición / el resto del espacio disponible menos lo que tenga calculado asignar para la partición de intercambio (200% de la memoria física, o cuanto baste para 2 GB). Se recomienda asignar / como partición primaria, siempre que la tabla de particiones lo permita.



Si está conforme, haga clic otra vez en el botón «Nuevo» y proceda a crear la siguiente partición.

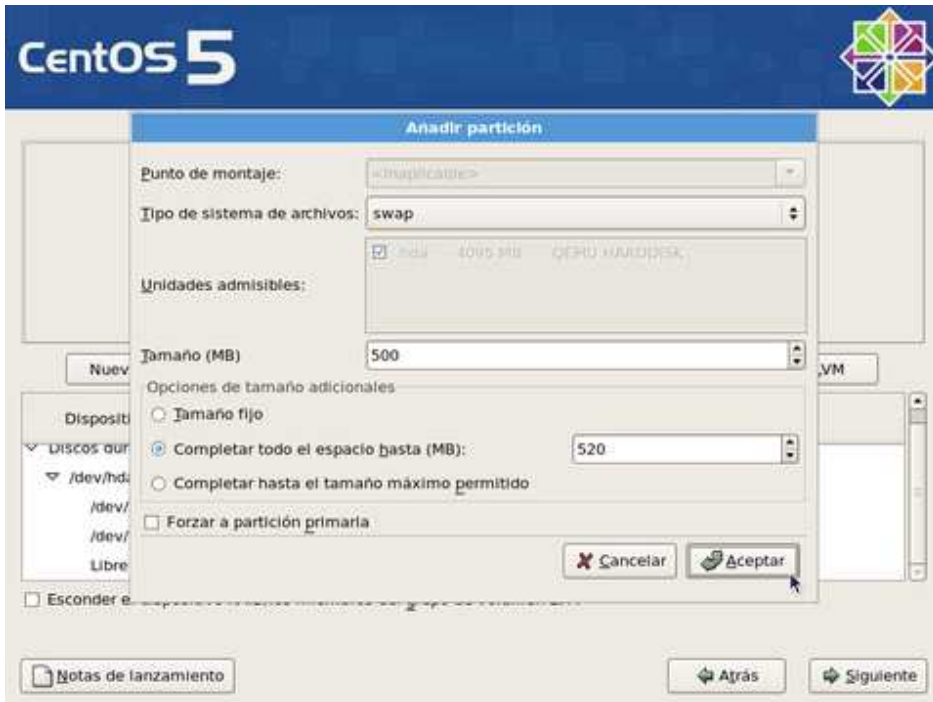


La partición para la memoria de intercambio no requiere punto de montaje. Seleccione en el campo de «Tipo de sistema de archivos» la opción «swap», asigne el 200% de la memoria física (o cuanto basta para 2 GB).

Por tratarse de la última partición de la tabla, es buena idea asignarle el espacio por rango, especificando valores ligeramente por debajo y ligeramente por arriba de lo planeado.

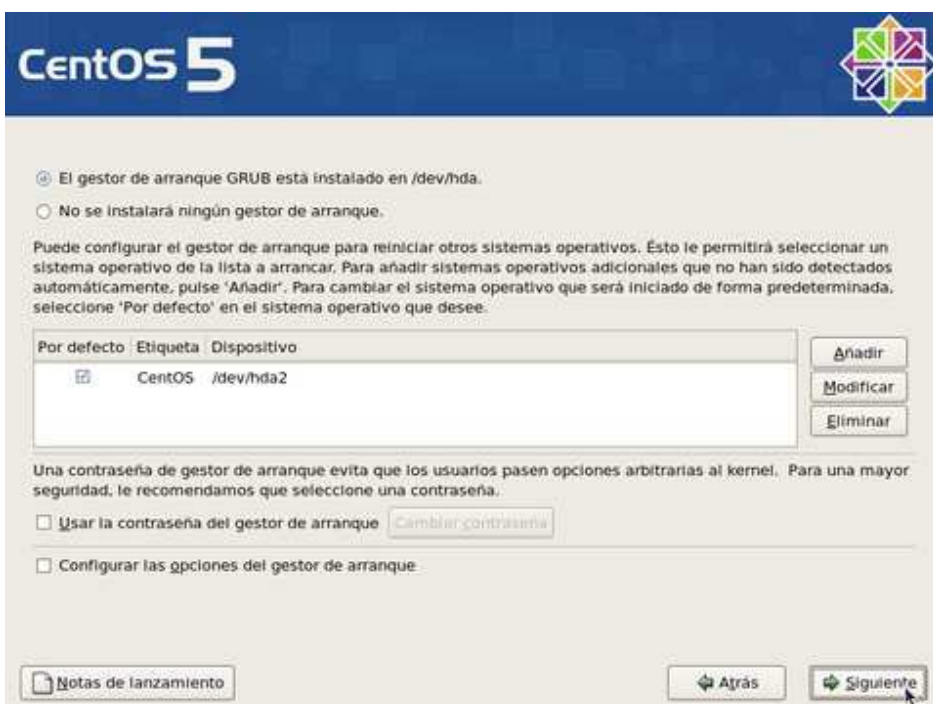
Otras particiones que se recomienda asignar, si se dispone del espacio en disco duro suficiente, son:

/usr	Requiere al menos 1.5 GB en instalaciones básicas. Debe considerarse el sustento lógico a utilizar a futuro. Para uso general, se recomiendan no menos de 5 GB y, de ser posible, considere un tamaño óptimo de hasta 8 GB en instalaciones promedio.
/tmp	Requiere al menos 350 MB y puede asignarse hasta 2 GB o más dependiendo de la carga de trabajo y tipo de aplicaciones. Si por ejemplo el sistema cuenta con un grabador de DVD, será necesario asignar a /tmp el espacio suficiente para almacenar una imagen de disco DVD, es decir, al menos 4.2 GB.
/var	Requiere al menos 512 MB en estaciones de trabajo sin servicios. En servidores regularmente se le asigna al menos la mitad del disco duro.
/home	En estaciones de trabajo se asigna al menos la mitad del disco duro a esta partición.

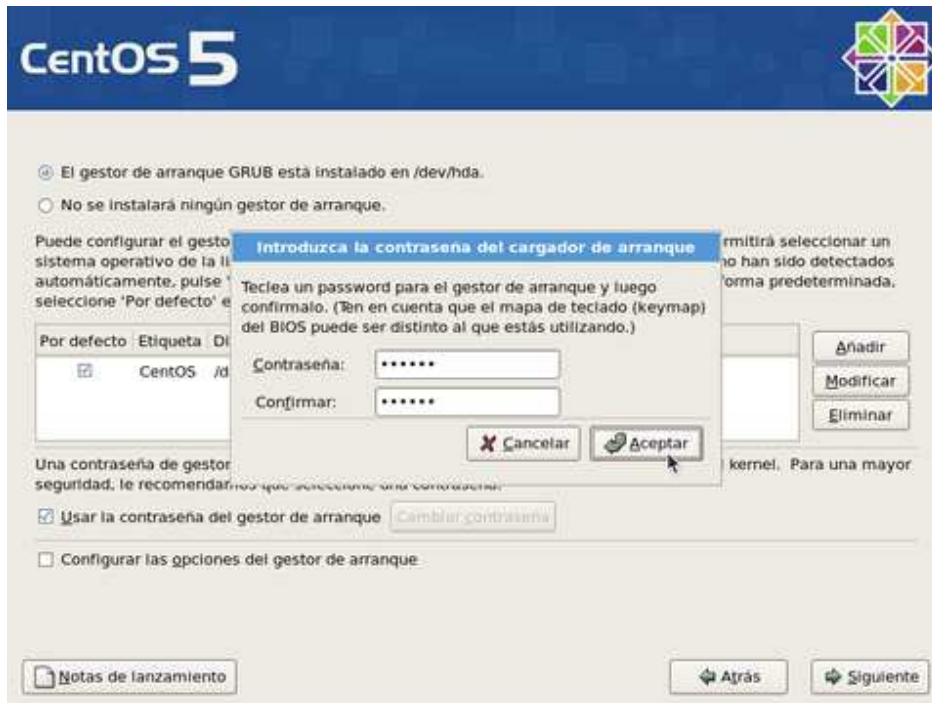


Si está conforme con la tabla de particiones creada, haga clic sobre el botón «siguiente» para pasar a la siguiente pantalla.

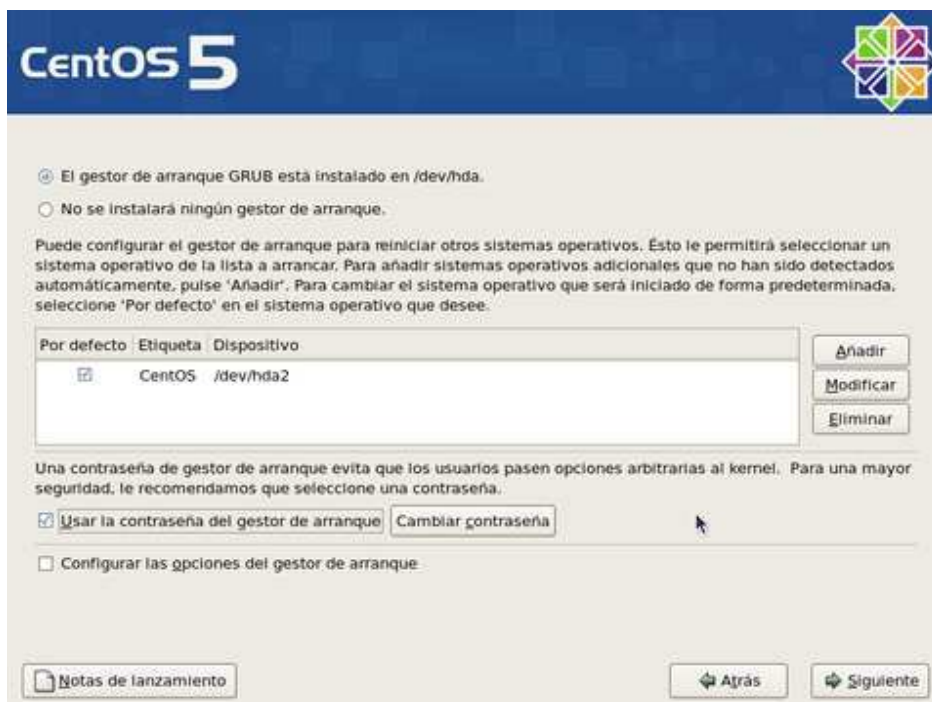
Ingresará a la configuración del gestor de arranque. Por motivos de seguridad, y principalmente con la finalidad de impedir que alguien sin autorización y con acceso físico al sistema pueda iniciar el sistema en nivel de corrida 1, o cualquiera otro, haga clic en la casilla «Usar la contraseña del gestor de arranque».



Se abrirá una ventana emergente donde deberá ingresar, con confirmación, la clave de acceso exclusiva para el gestor de arranque. Al terminar, haga clic sobre el botón «Aceptar».



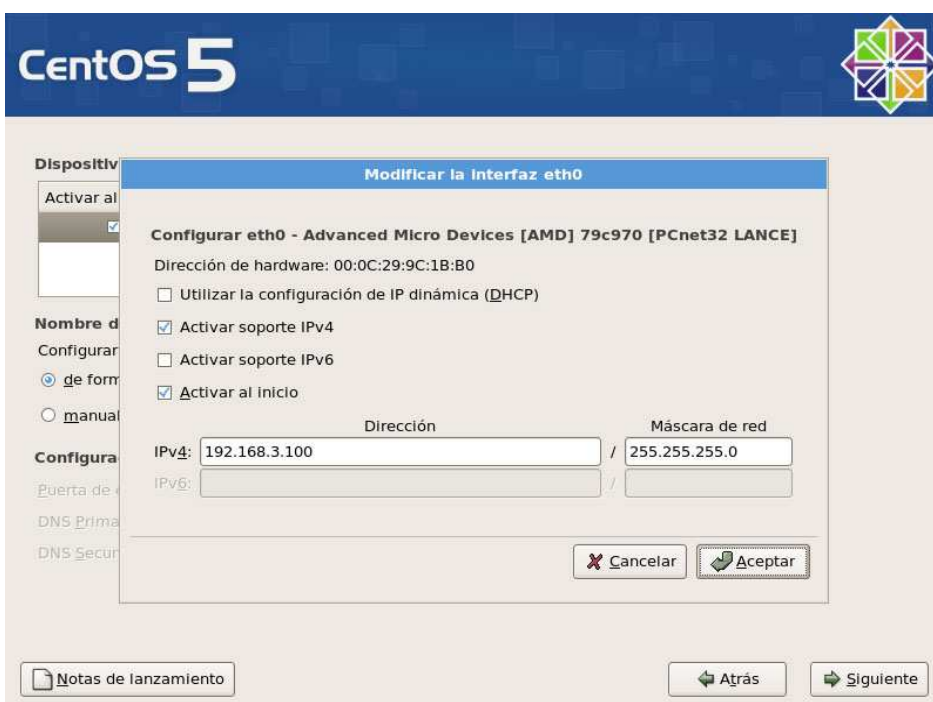
Al terminar, haga clic sobre el botón «Siguiente».



Para configurar los parámetros de red del sistema, haga clic sobre el botón «Modificar» para la interfaz eth0.



En la ventana emergente para modificar la interfaz eth0, desactive la casilla «Configurar usando DHCP» y especifique la dirección IP y máscara de subred que utilizará en adelante el sistema. Si no va a utilizar IPv6, también desactive la casilla. Confirme con el administrador de la red donde se localice que estos datos sean correctos antes de continuar. Al terminar, haga clic sobre el botón «Aceptar».



Asigne un nombre de anfitrión (HOSTNAME) para el sistema. Se recomienda que dicho nombre sea un FQDN (Fully Qualified Domain Name) resuelto al menos en un DNS local. Defina, además, en esta misma pantalla, la dirección IP de la puerta de enlace y las direcciones IP de los servidores DNS de los que disponga.

Si desconoce que dato ingresar, defina éste como localhost.localdomain. Al terminar, haga clic sobre el botón «Siguiente».



CentOS 5

Dispositivos de red

Activar al inicio	Dispositivo	IPv4/Máscara de red	IPv6/Prefijo	Modificar
<input checked="" type="checkbox"/>	eth0	192.168.3.100/24	Desactivado	

Nombre del Host

Configurar el nombre del host:

de forma automática a través de DHCP

manualmente (ej. "mipc.dominio.com.ar")

Configuración miscelánea

Puerta de enlace:

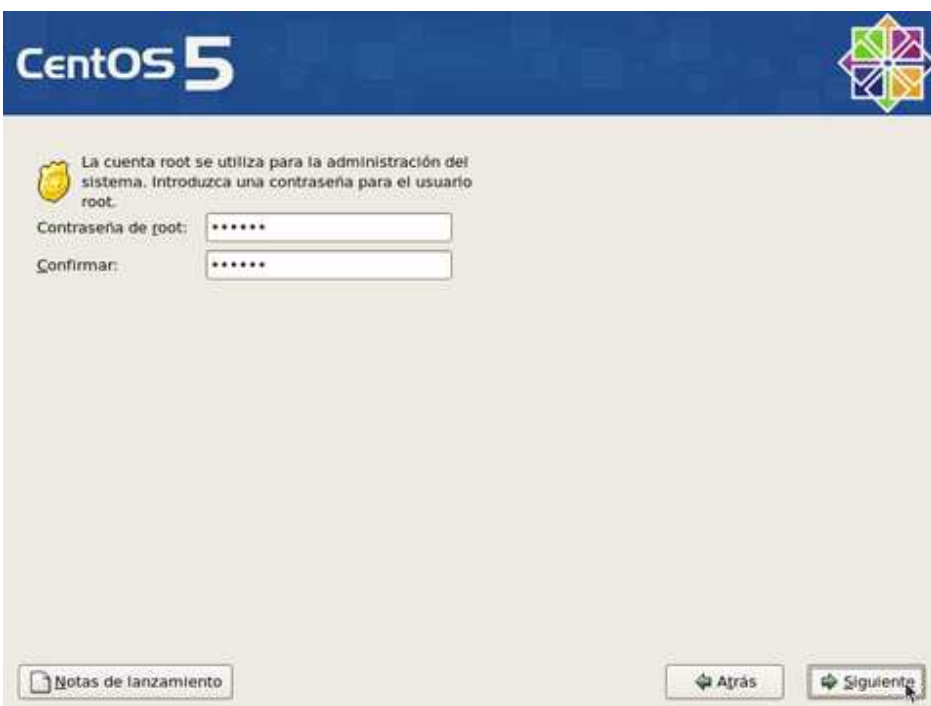
DNS Primario:

DNS Secundario:

Seleccione la casilla «El sistema horario usará UTC», que significa que el reloj del sistema utilizará UTC (Tiempo Universal Coordinado), que es el sucesor de GMT (Greenwich Mean Time, que significa Tiempo Promedio de Greenwich), y es la zona horaria de referencia respecto a la cual se calculan todas las otras zonas del mundo. Haga clic con el ratón sobre la región que corresponda en el mapa mundial o seleccione en el siguiente campo la zona horaria que corresponda a la región donde se hospedarán físicamente el sistema.



Asigne una clave de acceso al usuario root. Debe escribirla dos veces a fin de verificar que está coincide con lo que realmente se espera. Por razones de seguridad, se recomienda asignar una clave de acceso que evite utilizar palabras provenientes de cualquier diccionario, en cualquier idioma, así como cualquier combinación que tenga relación con datos personales.

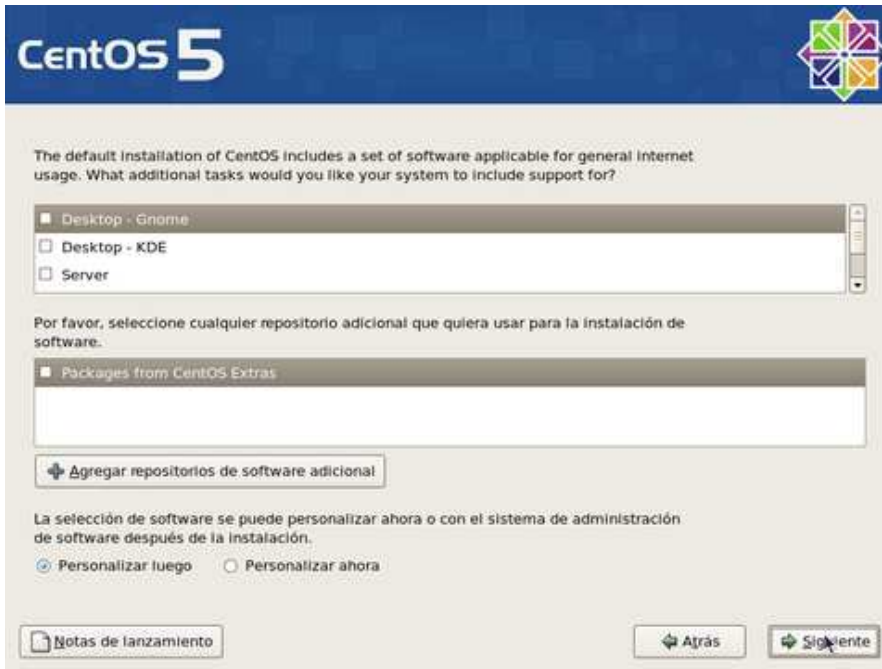


Al terminar, haga clic sobre el botón «Siguiente», y espere a que el sistema haga la lectura de información de los grupos de paquetes.



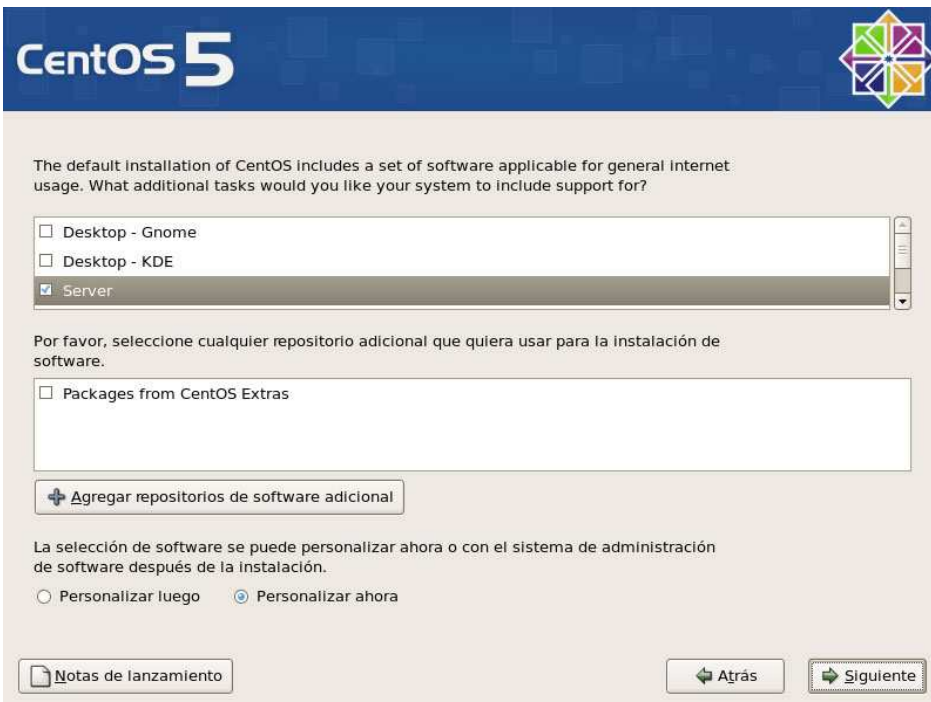
En la siguiente pantalla podrá seleccionar los grupos de paquetes que quiera instalar en el sistema. Añada o elimine a su conveniencia. Lo recomendado, sobre todo si se trata de un servidor, es realizar una instalación con el mínimo de paquetes, desactivando todas las casillas para todos los grupos de paquetes.

El objeto de esto es solo instalar lo mínimo necesario para el funcionamiento del sistema operativo, y permitir instalar posteriormente solo aquello que realmente se requiera de acuerdo a la finalidad productiva que tendrá el sistema. Al terminar, haga clic sobre el botón «Siguiente».



Se realizará una comprobación de dependencias de los paquetes a instalar. Este proceso puede demorar algunos minutos.



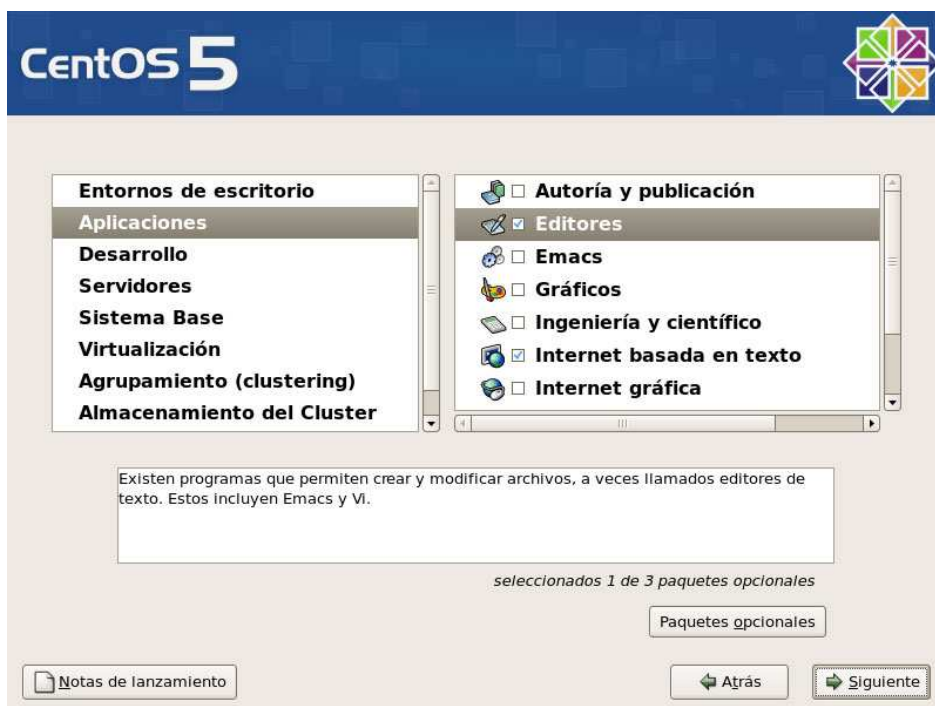


Ahora deberemos seleccionar el grupo de paquetes que queremos instalar.

Aplicaciones: Editores, Internet Basado en Texto, Internet gráfica, Oficina/Productividad.

Desarrollo: bibliotecas de desarrollo, Herramientas de Desarrollo.

Servidores: Base de datos MySQL, Herramientas para configuración del Servidor, Servidor Web, Servidor DNS.



Antes de iniciar la instalación sobre el disco duro, el sistema le informará respecto a que se guardará un registro del proceso en si en el fichero /root/install.log. Para continuar, haga clic sobre el botón «Siguiente».



Si iniciará de forma automática el proceso de formato de las particiones que haya creado para instalar el sistema operativo. Dependiendo de la capacidad del disco duro, este proceso puede demorar algunos minutos.



Se realizará automáticamente una copia de la imagen del programa de instalación sobre el disco duro a fin de hacer más eficiente el proceso. Dependiendo de la capacidad del microprocesador y cantidad de memoria disponible en el sistema, este proceso puede demorar algunos minutos.



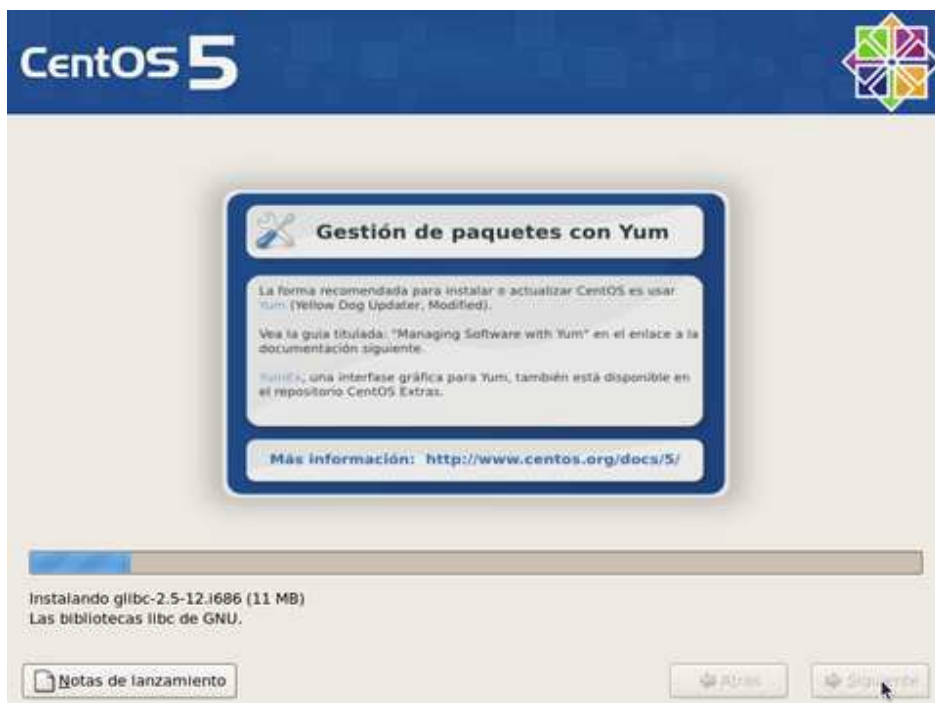
Espere a que se terminen los preparativos de inicio del proceso de instalación.



Se realizarán preparativos para realizar las transacciones de instalación de paquetes.



Iniciará la instalación de los paquetes necesarios para el funcionamiento del sistema operativo. Espere algunos minutos hasta que concluya el proceso.

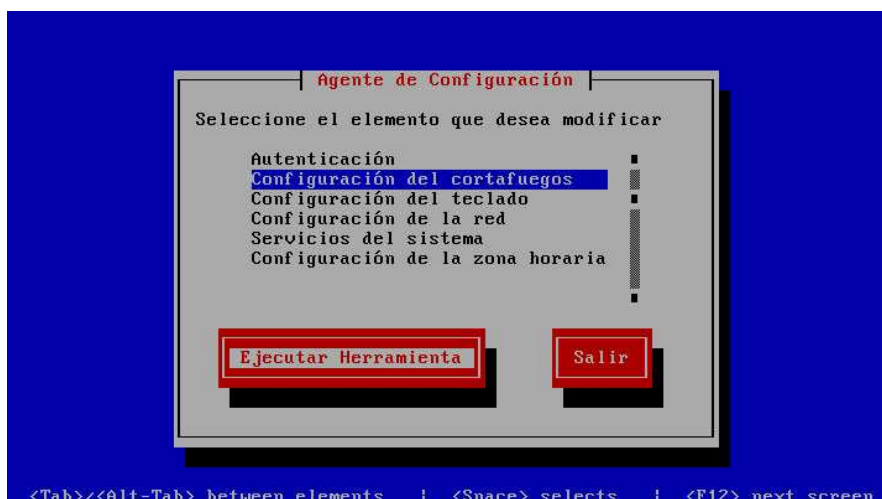


Una vez concluida la instalación de los paquetes, haga clic sobre el botón «Reiniciar».



3.10.5 CONFIGURACION CENTOS

Haciendo referencia al anterior paso, una vez reiniciada la máquina se está en la posibilidad de configurar el firewall. Seleccionamos Configuración del cortafuegos y presionamos ejecutar herramienta.



Se deshabilita el firewall Default de Centos en este momento. Se lo puede dejar para configurarlo después, esto es aconsejable, ya que podría interferir con la instalación de un firewall diferente del de CENTOS.

SELinux es una extensión de seguridad del CENTOS que podría proveer una seguridad más extensa, la deshabilitamos, porque existen muchas aplicaciones que dejarían de trabajar si la seguridad está habilitada.



Luego de esto abandonamos el Agente de Configuración, seleccionando salir. Entonces hacemos login como root y reiniciamos el sistema, para que los cambios sean aplicados:

```
shutdown -r now
```

Continuamos configurando el fichero hosts. Seguido editamos */etc/hosts*:

```
vi /etc/hosts
```

Quedando de la siguiente manera:

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
192.168.3.100     sayce.com.ec sayce
::1                localhost6.localdomain6 localhost6
```

Ahora comprobamos que el firewall y SELinux estén deshabilitados, poniendo en la línea de comandos lo siguiente:

```
system-config-securitylevel
```

Deshabilite el Nivel de seguridad y el SELinux.



Luego deberemos reiniciar el sistema.

```
shutdown -r now
```

Instalamos software adicional.

Importamos el GPG keys para paquetes de software:

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY*
```

Entonces hacemos una actualización de los paquetes existentes en el sistema:

```
yum update
```

Ahora se instalará algunos paquetes de software que serán necesarios después:

```
yum install wget bzip2 unzip zip nmap openssl lynx fileutils ncftp gcc gcc-c++
```

3.10.6 CONFIGURACION MySQL

Para instalar MySQL, deberemos hacer esto:

```
yum install mysql mysql-devel mysql-server
```

Luego se debe crear en el sistema un link de inicio para MySQL(esto hará que MySQL arranque automáticamente cada vez que el sistema inicie) entonces se iniciará el servidor de MySQL:

```
chkconfig --levels 235 mysqld on
```

```
/etc/init.d/mysqld start
```

Ahora verifiquemos que la red esté habilitada ejecutando:

```
netstat -tap | grep mysql
```

Lo cual mostraría la siguiente línea:

```
[root@sayce named]# netstat -tap | grep mysql
```

```
tcp    0    0 *:mysql          *.*              LISTEN    2470/mysqld
```

```
[root@sayce named]#
```

Si no se realizó lo anterior, editar `/etc/my.cnf` y **ponemos como comentario la línea `skip-networking`**:

```
vi /etc/my.cnf
```

```
#skip-networking
```

Se reiniciará el servidor de MySQL:

```
/etc/init.d/mysqld restart
```

Ejecutamos: la siguiente línea para asignar un password al usuario root (de otra manera cualquiera podría ingresar a la base de datos MySQL).

```
mysqladmin -u root password aquivaelpassword
```

```
mysqladmin -h sayce.com.ec -u root password aquivaelpassword
```

3.10.7 CONFIGURACION APACHE Y PHP

Se procederá a instalar APACHE con PHP:

```
yum install php php-devel php-gd php-imap php-ldap php-mysql php-odbc php-pear php-xml php-xmlrpc curl curl-devel perl-libwww-perl ImageMagick
```

```
libxml2 libxml2-devel
```

Luego editamos: `/etc/httpd/conf/httpd.conf`:

```
vi /etc/httpd/conf/httpd.conf
```

Y se cambiará el `DirectoryIndex` **por lo siguiente**:

```
[...]
```

```
DirectoryIndex index.html index.htm index.shtml index.cgi index.php index.php3  
index.pl
```

```
[...]
```

Ahora se configurará para que el sistema arranque con APACHE al mismo tiempo:

```
chkconfig --levels 235 httpd on
```

Iniciamos el Servidor de Apache

```
/etc/init.d/httpd start
```


CAPITULO 4

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Se recomienda el uso de esta Metodología, ya que es una de las más modernas de entre las metodologías XP, además que el contenido de su obra facilita la ejecución de un proyecto orientado a la Web. El autor de esta metodología enfatiza que en ciertos ambientes de desarrollo no es obligatorio realizar todos los diagramas presentados por la herramienta de diagramación UML, sino realizar aquellos diagramas que son útiles para la comprensión del equipo del proyecto y así lograr el producto deseado. Por lo que algunos diagramas clásicos entorpecen el entendimiento del proyecto.
- Se recomienda la instalación de los proyectos Web en Linux Centos por su solidez de los servicios. La integración del servidor Web Apache con el servidor de Base de datos MySql y el lenguaje de programación PHP y su compilador hacen que Linux (LAMP) garantice rendimiento y la estabilidad.
- Por el costo que significa la administración de un sitio WEB con otras herramientas respecto a Linux, se recomienda utilizar las antes indicadas por ser gratuitas y de fácil acceso.

4.2 RECOMENDACIONES

- Antes de realizar un proyecto informático de desarrollo de software se recomienda analizar la metodología que mas se ajuste a las necesidades del negocio, que resulte de manejo sencillo y útil para lograr un ágil y rápido desarrollo, tal como promulgan las últimas metodologías que son XP y Agile.
- Se recomienda tener previo conocimiento sobre las herramientas para el desarrollo del proyecto. Esto beneficiará en la rápida codificación y evitará un retraso al tener que realizar un aprendizaje de la herramienta.

- Se recomienda que los futuros desarrollos que se realicen en el lenguaje PHP se las realice con versiones PHP 5 en adelante, ya que a partir de esta versión el compilador y su estructura soporta programación orientada a objetos.

4.3 Bibliografía:

- Beck, K. (1999). Embracing change with extreme programming. *IEEE Computer*, 32(10), 70-77.
- Beck, K. (2000). *Extreme programming explained: Embrace change*. Addison-Wesley Professional.
- Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The unified modeling language user guide*. Addison-Wesley Professional.
- Cockburn, A. (2001a). *Writing effective use case*. ACM Press.
- Trowbridge David (2003). *Enterprise Solution Patterns Using Microsoft.NET*. Microsoft
- Favre, L. (Ed.). (2003). *UML and the unified process*. IRM Press.
- Fioravanti, Fabrizio. (2003). The impact of eXtreme programming on maintenance. In M. Polo, M. Piattini, & F. Ruiz (Eds.), *Advances in software maintenance management: Technologies and solutions* (pp. 75-92). Idea Group Publishing.
- Fioravanti, Fabrizio. (2006). *Skills for Managing Rapidly Changing IT Projects*. IRM Press.
- Fowler, M. (2004). *UML distilled : A brief guide to the standard object modeling language* (3rd ed.). Addison-Wesley.
- Larman, C. (2001). *Applying UML and patterns: An introduction to object-oriented analysis and design and the unified process*. Prentice Hall.
- Mellor, S., Balcer, M., & Balcer, M. J. (2002). *Executable UML: A foundation for model driven architecture*. Addison-Wesley.
- Paulk, M. C., Curtis, B. Chrissis, M. B., & Weber, C. (1993, February). *Capability maturity model for software, Version 1.1* (Publication CMU/SEI-93-TR-24, DTIC No. ADA263403). Software Engineering Institute.
- Paulk, M. C., Weber, C. V., Garcia, S. M., Chrissis, M. B., & Bush, M. W. (1993, February). *Key practices of the capability maturity model, Version 1.1* (Publication CMU/SEI-93-TR-25, DTIC No. ADA263432). Software Engineering Institute.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The unified modeling language reference manual*. Boston, MA: Addison-Wesley.

Adobe (Agosto 2008). *Características de la herramienta para animación Web*. Disponible en: <http://www.adobe.com/es/products/flash/features/>

Adobe (Agosto 2008). *Características de la herramienta para edición Web*. Disponible en: <http://www.adobe.com/es/products/dreamweaver/features/>

Bulma (Septiembre 2006). *Guía de instalación de LAMP*. Disponible en: <http://bulmalug.net/body.phtml?nIdNoticia=628>

CentOS (Septiembre 2006). *Sistema operativo Centos homepage*. Disponible en: <http://www.centos.org>

HowtoForge (Septiembre 2006). *Guía de instalación de Linux Centos*. Disponible en: <http://www.howtoforge.com/installation-guide-centos5.1-desktop/>

IBM-Rational Software (Septiembre 2006). *Herramienta para Análisis y Diseño de Software*. Disponible en: <http://www.rational.com.ar/herramientas/rationalsystemsdeveloper.html>

OMG (Agosto 2008). *OMG home page*. Disponible en : <http://www.omg.org>

PHP (Septiembre 2006). *Manual de PHP*. Disponible en: <http://www.php-es.com/>

RPMS (Noviembre 2006). *RPMS para Linux*. Disponible en: <http://rpms.arvin.dk/>

UML (Agosto 2008). *UML homepage*. Disponible en: <http://www.uml.org>

WebEstilo (Octubre 2006). *Manuales de PHP*. Disponible en: <http://webestilo.com/php/>

ANEXO A:

MANUAL DEL USUARIO

CONTENIDO DEL MANUAL DEL USUARIO

1. CONTENIDO WEB DE LA SAYCE	94
1.1 PÁGINA DE INICIO: INDEX	94
1.2 BARRA PRINCIPAL: SAYCE-0.SWF	95
1.3 PÁGINA SALUDO INICIAL	96
1.4. PÁGINA QUIENES SOMOS	97
1.5 BASES LEGALES	98
1.5.1 BASES LEGALES: ACTA CONSTITUTIVA	99
1.5.2 BASES LEGALES: ESTATUTOS DE LA SAYCE	100
1.5.3 DERECHOS DE AUTOR	101
1.5.4 LEY DE PROPIEDAD INTELECTUAL	102
1.5.5 CONVENIO DE BERNA	103
1.5.6 CONVENIO DE ROMA	104
1.5.7 ACUERDOS ADPIC	105
1.6. GALERIA DE FOTOS	106
1.7. SOCIOS	107
1.8. USUARIOS Y TARIFAS	108
1.8.1 TARIFAS POR REPRODUCCION	109
1.8.2 DISTRIBUCION	110
1.8.3 SINCRONIZACION	111
1.8.4 COMUNICACIÓN PÚBLICA	112
1.9 NOTICIAS	113
1.10 WEBMAIL	114
1.11 BUSQUEDA	115
1.11.1 BUSQUEDA DE OBRAS	116
1.11.2 BUSQUEDA DE AUTORES	118
1.11.3 BUSQUEDA DE BIOGRAFIA	120
2. ADMINISTRADOR DE CATÁLOGOS DE AUTORES/COMPOSITORES, OBRAS, BIOGRAFÍAS	122
2.1 INGRESO AL ADMINISTRADOR	122
2.2 MENU DEL ADMINISTRADOR	123
2.3 ADMINISTRADOR DE OBRAS	124
2.3.1 CREANDO OBRA	125
2.3.2 EDITANDO OBRA	126
2.3.3 EXCLUIR UN SOCIO DE UNA OBRA	129
2.4 ADMINISTRADOR DE AUTORES	130
2.4.1 CREANDO SOCIO	131
2.4.2 EDITANDO SOCIO	132
2.4.3 AGREGANDO OBRA AL SOCIO	133
2.4.3 EXCLUIR OBRA DE UN SOCIO	134
2.4.4 ELIMINANDO UN SOCIO	135
2.5 ADMINISTRADOR DE BIOGRAFIAS	136
2.5.1 DETALLE DE BIOGRAFIA	137
2.5.2 EDITANDO BIOGRAFIA	138
2.5.3 CREANDO NUEVA BIOGRAFIA	140
2.5.4. ELIMINANDO BIOGRAFIAS	141

1. CONTENIDO WEB DE LA SAYCE

1.1 PÁGINA DE INICIO: INDEX



<p>DESCRIPCIÓN PÁGINA: INICIO</p> <p>El diseño de la página tiene una animación en flash. Que contiene las fotos de los autores nacionales más connotados de la SAYCE, así como su LOGO y Emblema y la opción SALTAR INTRO para ir a la página principal.</p>	<p>index.html UBICACIÓN: Sayce\flash\html LINKS Link: sayce-0.swf</p>
--	---

1.2 BARRA PRINCIPAL: SAYCE-0.SWF



<p>DESCRIPCIÓN BARRA PRINCIPAL</p> <p>El diseño de ésta barra esta en flash, consta de: la parte superior contiene el LOGO y un Menú que permite el acceso a otras páginas, esta página se mantiene presente en todas las páginas de la WEB.</p>	<p>sayce-0.swf UBICACIÓN: Sayce\flash\html LINKS</p> <p>Link: PAGINA PRINCIPAL.SWF Link: QUIENES SOMOS.SWF Link: BASES LEGALES.SWF Link: GALERIA DE FOTOS.SWF Link: SOCIOS.SWF Link: USUARIOS-TARIFAS.SWF Link: NOTICIAS.SWF Link: REVISTA.SWF Link: BUSCAR AUTOR .PHP Link: BUSCAR OBRA.PHP Link: BUSCAR BIOGRAFIA .PHP Link: NUESTRA MUSICA.SWF</p>
---	--

1.3 PÁGINA SALUDO INICIAL

SOCIEDAD DE AUTORES Y COMPOSITORES ECUATORIANOS

Visión y Misión CISAC Y LATINAUTOR Plan Estratégico Contáctenos

[PAGINA PRINCIPAL](#)
[QUIENES SOMOS](#)
[BASES LEGALES](#)
[GALERIA DE FOTOS](#)
[SOCIOS](#)
[USUARIOS-TARIFAS](#)
[NOTICIAS](#)
[REVISTA](#)
[BUSCAR AUTOR/COMPOSITOR](#)
[BUSCAR OBRA](#)
[BUSCAR BIOGRAFÍA](#)
[NUESTRA MUSICA](#)

2008

→ Visión
 Alcanzar el prestigio nacional e internacional como una sociedad de gestión colectiva de Derechos de Autor, gestionando efizcamente los derechos económicos de autor, con una óptima aplicación de procesos, en la administración de los repertorios representados por la SAYCE.

→ Misión
 Proteger, Recaudar y Distribuir los Derechos de Autor, de las obras de nuestros socios nacionales y extranjeros, a través de los Contratos de Reciprocidad, fomentando una conciencia colectiva sobre la propiedad intelectual, mejorando la calidad de vida y servicio institucional.

DR. MANUEL RIERA
DIRECTOR GENERAL

Visitando: Página Principal SAYCE © 2008 - SAYCEinformática

DESCRIPCIÓN PÁGINA: PRINCIPAL

El diseño de esta página está en flash. Contiene: botones en flash que despliegan información como: la Visión y Misión, CISAC LATINAUTOR, Plan Estratégico, Contáctenos.

sayce-1.swf

UBICACIÓN: Sayce\flash\html

LINKS

Link: VISION Y MISION.SWF

Link: CISAC Y LATINAUTOR.SWF

Link: PLAN ESTRATEGICO.SWF

Link: CONTACTENOS.SWF

1.4. PÁGINA QUIENES SOMOS

SOCIEDAD DE AUTORES Y COMPOSITORES ECUATORIANOS

Que es la SAYCE? Fines de la SAYCE

PAGINA PRINCIPAL
QUIENES SOMOS
BASES LEGALES
GALERIA DE FOTOS
SOCIOS
USUARIOS-TARIFAS
NOTICIAS
REVISTA
BUSCAR AUTOR/COMPOSITOR
BUSCAR OBRA
BUSCAR BIOGRAFÍA
NUESTRA MUSICA

2008

E-MAIL

Visitando: Quienes somos

SAYCE © 2008 - SAYCEinformática

QUE ES LA SAYCE?

La Sociedad de Autores y Compositores Ecuatorianos, es una entidad de gestión colectiva, de derecho privado, con finalidad social; cuyo deber primordial es preservar los derechos morales y administrar los derechos económicos resultantes de la utilización pública de las obras de autores nacionales y/o extranjeros a lo que representan.

La ley de Derechos de Autor No. 610, publicada en R.O. No. 149 del 13-08-77, permitió que la "S.A.Y.C.E" adquiriera personería jurídica mediante Acuerdo Ministerial No.755, publicado el 28 de Enero de 1977.

Como Entidad Autoral, es miembro ordinario de la Confederación Internacional de Sociedades de Autores y Compositores, C.I.S.A.C., organismo intercontinental para la protección de Derechos de Autorales y de otros organismos regionales afines.

DESCRIPCIÓN PÁGINA: QUIENES SOMOS

El diseño de esta página está en flash, consta de dos botones que describen que es la SAYCE y sus fines.

sayce-2.swf

UBICACIÓN: Sayce\flash\html

LINKS

Link: QUE ES LA SAYCE.SWF

Link: FINES DE LA SAYCE.SWF

1.5 BASES LEGALES

SOCIEDAD DE AUTORES Y COMPOSITORES ECUATORIANOS

Bases Legales | Area Jurídica | Resoluciones | Datos Relevantes

- PAGINA PRINCIPAL
- QUIENES SOMOS
- BASES LEGALES
- GALERIA DE FOTOS
- SOCIOS
- USUARIOS-TARIFAS
- NOTICIAS
- REVISTA
- BUSCAR AUTOR/COMPOSITOR
- BUSCAR OBRA
- BUSCAR BIOGRAFÍA
- NUESTRA MUSICA

2008

ACTA CONSTITUTIVA
ESTATUTOS
DERECHO DE AUTOR
LEY DE PROPIEDAD INTELECTUAL
CONVENIO DE BERNA
CONVENIO DE ROMA
ACUERDOS ADPIC

Visitando: Bases Legales | SAYCE © 2008 - SAYCEinformática

DESCRIPCIÓN PÁGINA: BASES LEGALES

El diseño de esta página está en flash, contiene links a: Acta Constitutiva, Estatutos, Derechos de Autor, Ley de Propiedad Intelectual, Convenio de Berna, Convenio de Roma, Acuerdos ADPIC, así como también consta de tres botones que describen: el Area Jurídica, Resoluciones importantes de esta área y datos relevantes en el ámbito jurídico.

sayce-3.swf

UBICACIÓN: Sayce\flash\html

LINKS

Link: ACTA CONSTITUTIVA.HTML
Link: ESTATUTOS.HTML
Link: DERECHOS DE AUTOR.HTML
Link: LEY DE PROPIEDAD.HTML
Link: CONVENIO DE BERNA.HTML
Link: CONVENIO DE ROMA.HTML
Link: ACUERDOS ADPIC.HTML
Link: AREA JURIDICA.SWF
Link: RESOLUCIONES.SWF
Link: DATOS RELEVANTES.SWF

1.5.1 BASES LEGALES: ACTA CONSTITUTIVA



<p>DESCRIPCIÓN PÁGINA: ACTA CONSTITUTIVA</p> <p>El diseño de esta página está en html: Describe la historia de la SAYCE, así como su Acta Constitutiva.</p>	<p>sayce-3-bl1.html UBICACIÓN: Sayce\flash\html</p>
--	---

1.5.2 BASES LEGALES: ESTATUTOS DE LA SAYCE



DESCRIPCIÓN PÁGINA: ESTATUTOS

El diseño de esta página está en html,
contiene los Estatutos de la Sociedad.

sayce-3-bl2.html
UBICACIÓN: Sayce\flash\html

1.5.3 DERECHOS DE AUTOR



<p>DESCRIPCIÓN PÁGINA: DERECHOS DE AUTOR</p> <p>El diseño de esta página está en html, contiene todo lo concerniente a Derechos de Autor y Derechos Conexos.</p>	<p>sayce-3-bl3.html UBICACIÓN: Sayce\flash\html</p>
---	---

1.5.4 LEY DE PROPIEDAD INTELECTUAL

SAYCE Un emblema de honor...
ECUADOR ...Por el derecho de autor.

Registro Oficial No 320

Ley de Propiedad Intelectual

CONGRESO NACIONAL
 EL PLENARIO DE LAS COMISIONES LEGISLATIVAS

Considerando:
 Que la protección de las creaciones intelectuales es un derecho fundamental, así concebido en la Declaración Universal de los Derechos Humanos, aprobada por la Asamblea General de la ONU en 1948; Que es función del Estado asumir la defensa de los derechos intelectuales, que la protección de la propiedad intelectual es vital para el desarrollo tecnológico y económico del País, fomenta inversión en investigación y desarrollo, estimula la producción tecnológica nacional y confiere al Ecuador una ventaja comparativa en el nuevo orden económico mundial; Que la falta de una adecuada protección a los derechos de propiedad intelectual restringe la libre competencia y obstaculiza el crecimiento económico respecto de la más amplia gama de bienes y servicios que incorporan activos intangibles; Que la competitividad de la industria y el comercio ecuatorianos en el mercado internacional depende cada vez más de su capacidad de incorporar avances tecnológicos a la producción y comercialización de sus bienes y servicios; Que la protección de los derechos intelectuales debe responder a los principios de universalidad y armonización internacional; Que el Ecuador se ha adherido a la Organización Mundial de Comercio y ha ratificado el Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio (ADPIC); Que están vigentes en el Ecuador varias normas de aplicación internacional que implican una reformulación integral de la legislación en materia de Propiedad Intelectual, como la protección a los derechos de autor, especialmente el Convenio de Berna para la Protección de Obras Literarias y Artísticas, Acta de París, la Convención de Roma sobre la Protección de los Artistas; Interpretes o Ejecutantes, los Productores de Fonogramas y los Organismos de Radiodifusión, que a pesar de su ratificación en 1983 no fue reflejada en nuestra legislación, la Convención Universal sobre Derechos de Autor, el Régimen Común sobre Derechos de Autor y Derechos Conexos, regulado en la Decisión N° 351 de la Comisión del Acuerdo de Cartagena, vigente para todos los países de la Comunidad Andina; y, la protección a la Propiedad Intelectual; Que el Estado debe optimizar los recursos humanos, tecnológicos y económicos, unificando la aplicación administrativa de las leyes sobre Propiedad Industrial, Obtenciones Vegetales y Derechos de Autor; y, En ejercicio de sus facultades constitucionales y legales, expide la siguiente: Registro Oficial No 320

TITULO PRELIMINAR

Art1. El Estado reconoce, regula y garantiza la propiedad intelectual adquirida de conformidad con la ley, las Decisiones de la Comisión de la Comunidad Andina y los convenios internacionales vigentes en el Ecuador. La propiedad intelectual comprende:

<p>DESCRIPCIÓN PÁGINA: LEY DE PROPIEDAD INTELECTUAL</p> <p>El diseño de esta página está en html, contiene la Ley de Propiedad Intelectual sobre la cual se rige SAYCE.</p>	<p>sayce-3-bl4.html UBICACIÓN: Sayce\flash\html</p>
--	--

1.5.5 CONVENIO DE BERNA



Convenio de Berna

CONVENIO DE BERNA PARA LA PROTECCIÓN DE LAS OBRAS LITERARIAS Y ARTÍSTICAS

Los países de la Unión, animados por el mutuo deseo de proteger de modo más eficaz y uniforme posible los derechos de los autores sobre sus obras literarias y artísticas.

Reconociendo la importancia de los trabajos de la Conferencia de Revisión celebrada en Estocolmo en 1967.

Han resuelto revisar el Acta adoptada por la Conferencia de Estocolmo, manteniendo sin modificaciones los artículos 1 a 20 y 22 a 26 de esa Acta.

En consecuencia, los Plenipotenciarios que suscriben, luego de haber sido reconocidos y aceptados en debida forma los plenos poderes presentados, han convenido lo siguiente.

ARTICULO 1

Los países a los cuales se aplica el presente Convenio están constituidos en Unión para la protección de los derechos de los autores sobre sus obras literarias y artísticas.

ARTICULO 2

- 1) Los términos "obras literarias y acústicas" comprenden todas las producciones en el campo literario, científico y artístico, cualquiera que sea el modo o forma de expresión, tales como los libros, folletos y otros escritos; las conferencias, alocuciones, sermones y otras obras de la misma naturaleza, las obras dramáticas o dramático-musicales; las obras coreográficas y las pantomimas; las composiciones musicales con o sin letra; las obras cinematográficas, a las cuales se asimilan las obras expresadas por procedimiento análogo a la cinematografía, las obras de dibujo, pintura, arquitectura, escultura, grabado, litografía, las obras fotográficas a las cuales se asimilan las expresadas por procedimiento análogo a la fotografía; las obras de artes aplicadas; las ilustraciones, mapas, planos, croquis y obras plásticas relativas a la geografía, a la topografía, a la arquitectura o a las ciencias.
- 2) Sin embargo, queda reservada a las legislaciones de los países de la Unión la facultad de establecer que las obras literarias y artísticas o algunos de sus géneros no estarán protegidos mientras no hayan sido fijados en un soporte material.
- 3) Estarán protegidas como obras originales, sin perjuicio de los derechos del autor de la obra original, las traducciones, adaptaciones, arreglos musicales y demás transformaciones de una obra literaria o artística.
- 4) Queda reservada a las legislaciones de los países de la Unión la facultad de determinar la protección que han de conceder a los textos oficiales de orden legislativo, administrativo o judicial, así como a las traducciones oficiales de estos textos.
- 5) Las colecciones de obras literarias o artísticas tales como las enciclopedias y antologías que, por la selección o

DESCRIPCIÓN PÁGINA: CONVENIO DE BERNA	sayce-3-bl5.html UBICACIÓN: Sayce\flash\html
El diseño de esta página está en html, contiene el Convenio de Berna.	

1.5.6 CONVENIO DE ROMA



Convenio de Roma

LEY APROBATORIA DE LA CONVENCION INTERNACIONAL SOBRE LA PROTECCION DE LOS ARTISTAS INTERPRETES O EJECUTANTES, LOS PRODUCTORES DE FONOGRAMAS Y LOS ORGANISMOS DE RADIODIFUSION, SUSCRITA EN ROMA EL 26 DE OCTUBRE DE 1961

CONVENCION INTERNACIONAL SOBRE LA PROTECCION DE LOS ARTISTAS INTERPRETES O EJECUTANTES, LOS PRODUCTORES DE FONOGRAMAS Y LOS ORGANISMOS DE RADIODIFUSION

Hecho en Roma el 26 de octubre de 1961 Los Estados Contratantes, animados del deseo de asegurar la proteccion de los derechos de los artistas, intérpretes o ejecutantes, de los productores de fonogramas y de los organismos de radiodifusion.

Han convenido:

Artículo PRIMERO

La proteccion prevista en la presente Convencion dejará intacta y no afectará en modo alguno a la proteccion del derecho de autor sobre las obras literarias y artisticas. Por lo tanto, ninguna de las disposiciones de la presente Convencion podrá interpretarse en menoscabo de esa proteccion.

Artículo 2

1. A los efectos de la presente Convencion se entenderá por "mismo trato que a los nacionales" el que conceda el Estado Contratante en que se pida la proteccion, en virtud de su derecho interno:

- a) a los artistas, intérpretes o ejecutantes que sean nacionales de dicho Estado, con respecto a las interpretaciones o ejecuciones realizadas, fijadas por primera vez o radiodifundidas en su territorio,
- b) a los productores de fonogramas que sean nacionales de dicho Estado, con respecto a los fonogramas publicados o fijados por primera vez en su territorio,
- c) a los organismos de radiodifusion que tengan su domicilio legal en el territorio de dicho Estado, con respecto a las emisiones difundidas desde emisoras situadas en su territorio.

2. el "mismo trato que a los nacionales" estará sujeto a la proteccion expresamente concedida y a las limitaciones concretamente previstas en la presente Convencion.

Artículo 3

A los efectos de la presente Convencion, se entenderá por:

**DESCRIPCIÓN PÁGINA:
CONVENIO DE ROMA**

El diseño de esta página está en html,
contiene el Convenio de Roma.

sayce-3-bl6.html
UBICACIÓN: Sayce\flash\html

1.5.7 ACUERDOS ADPIC



Acuerdo sobre los ADPIC

ACUERDO SOBRE LOS ASPECTOS DE LOS DERECHOS DE PROPIEDAD INTELECTUAL RELACIONADOS CON EL COMERCIO (ACUERDO SOBRE LOS ADPIC) (1994)

Los Miembros,
Deseosos de reducir las distorsiones del comercio internacional y los obstáculos al mismo, y teniendo en cuenta la necesidad de fomentar una protección eficaz y adecuada de los derechos de propiedad intelectual y de asegurarse de que las medidas y procedimientos destinados a hacer respetar dichos derechos no se conviertan a su vez en obstáculos al comercio legítimo;

Reconociendo, para este fin, la necesidad de nuevas normas y disciplinas relativas a:

- a) la aplicabilidad de los principios básicos del GATT de 1994 y de los acuerdos o convenios internacionales pertinentes en materia de propiedad intelectual;
- b) la provisión de normas y principios adecuados relativos a la existencia, alcance y ejercicio de los derechos de propiedad intelectual relacionados con el comercio;
- c) la provisión de medios eficaces y apropiados para hacer respetar los derechos de propiedad intelectual relacionados con el comercio, tomando en consideración las diferencias entre los sistemas jurídicos nacionales;
- d) la provisión de procedimientos eficaces y ágiles para la prevención y solución multilaterales de las diferencias entre los gobiernos; y
- e) disposiciones transitorias encaminadas a conseguir la más plena participación en los resultados de las negociaciones,

Reconociendo la necesidad de un marco multilateral de principios, normas y disciplinas relacionados con el comercio internacional de mercancías falsificadas;

Reconociendo que los derechos de propiedad intelectual son derechos privados;

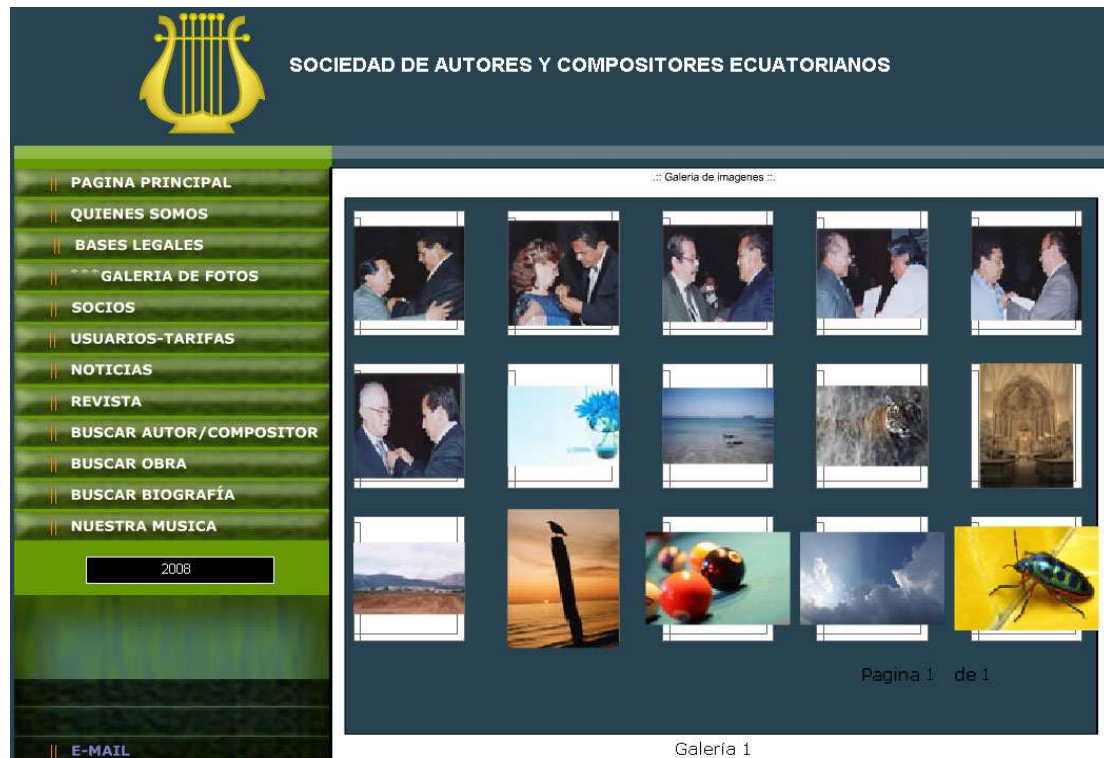
Reconociendo los objetivos fundamentales de política general pública de los sistemas nacionales de protección de los derechos de propiedad intelectual, con inclusión de los objetivos en materia de desarrollo y tecnología;

Reconociendo asimismo las necesidades especiales de los países menos adelantados Miembros por lo que se refiere a la aplicación, a nivel nacional, de las leyes y reglamentos con la máxima flexibilidad requerida para que esos países estén en condiciones de crear una base tecnológica sólida y viable;

Insistiendo en la importancia de reducir las tensiones mediante el logro de compromisos más firmes de resolver por medio de procedimientos multilaterales las diferencias sobre cuestiones de propiedad intelectual relacionadas con el comercio;

<p style="text-align: center;">DESCRIPCIÓN PÁGINA: ACUERDOS ADPIC</p> <p>El diseño de esta página está en html, describe los Acuerdos ADPIC.</p>	<p style="text-align: center;">sayce-3-bl7.html UBICACIÓN: Sayce\flash\html</p>
---	---

1.6. GALERIA DE FOTOS



DESCRIPCIÓN PÁGINA: GALERIA DE FOTOS

El diseño de esta página está en flash, contiene imágenes de los últimos eventos que ha tenido SAYCE, al momento de hacer clic sobre cualquiera de las imágenes podremos agrandarla.

sayce-4.swf
UBICACIÓN: Sayceflash\html

1.7. SOCIOS

SOCIETY OF AUTHORS AND COMPOSERS ECUATORIANOS

Admisiones Requisitos | Beneficios | Atención Personalizada | Fomento a la producción

PAGINA PRINCIPAL
QUIENES SOMOS
BASES LEGALES
GALERIA DE FOTOS
SOCIOS
USUARIOS-TARIFAS
NOTICIAS
REVISTA
BUSCAR AUTOR/COMPOSITOR
BUSCAR OBRA
BUSCAR BIOGRAFÍA
NUESTRA MUSICA

2008

ADMISIONES

Serán admitidos como SOCIOS NUEVOS de la SAYCE, los autores y/o compositores que libre y voluntariamente manifiesten en su solicitud, su deseo de pertenecer a la sociedad y reúnan los requerimientos legales, reglamentarios y estatutarios correspondientes.

Si usted es una persona natural o jurídica con obras del ingenio protegida por derechos de autor o es titular de los derechos de explotación de obras musicales.....

Regístrese en SAYCE !!!

Para registrarse en SAYCE solicite INFORMACIÓN en el Dpto. de atención al socio sobre los siguientes pasos:

Paso 1. - Solicitud de ingreso con la información del tipo o categoría de socio.

E-MAIL | Visitando: Socios | SAYCE © 2008 - SAYCEinformática

DESCRIPCIÓN PÁGINA: SOCIOS

El diseño de esta página está en flash contiene cuatro botones que describen: admisiones y requisitos, Beneficios que la sociedad entrega a sus socios, Atención personalizada y el Fomento a la producción.

sayce-5.swf

**UBICACIÓN: Sayce\flash\html
LINKS**

Link: ADMISIONESREQUISITOS.SWF

Link: BENEFICIOS.SWF

Link: ATENCION PERSONALIZADA.SWF

Link: FOMENTO A LA PRODUCCION.SWF

1.8. USUARIOS Y TARIFAS



DESCRIPCIÓN PÁGINA: USUARIOS Y TARIFAS

El diseño de esta página está en flash, contiene links para cada concepto de Recaudación que cobra SAYCE, esto es: Reproducción, Distribución, Sincronización y Comunicación Pública, así como también contiene 3 botones que describen las gestiones realizadas por Usuarios Generales, Espectáculos Públicos y Fonomecánicos.

sayce-6.swf UBICACIÓN: Sayce\flash\html LINKS

Link: REPRODUCCION.HTML
Link: DISTRIBUCION.HTML
Link: SINCRONIZACION.HTML
Link: COMUNICACIÓN PUBLICA.HTML
Link: USUARIOS GENERALES.SWF
Link: ESPECTACULOS PUBLICOS.SWF
Link: FONOMECHANICOS.SWF

1.8.1 TARIFAS POR REPRODUCCION



TARIFAS POR REPRODUCCIÓN

DERECHOS DE REPRODUCCIÓN

Por los derechos de reproducción de obras musicales, sea que el soporte se produzca en territorio nacional o se importe, se aplicará la tarifa del 12% sobre el precio de venta al público de cada soporte fonomecánico fabricado, para lo cual se tomará como referencia la media del valor señalado por vendedores y detallistas en el mercado.

La tarifa a cobrarse por derechos de reproducción de obras administradas en formato video se tendrá en cuenta el precio de venta al público menos el porcentaje del IVA y el costo del soporte, al valor resultante se aplicarán los siguientes porcentajes:

- Película cinematográfica 1.5%
- Película cinematográfica musical 3%
- Comedias musicales 4%
- Didácticos documentales 1%
- Científicos, microprogramas
- Realizaciones coreográficas 2%
- Video clip 5%
- Show musical 6%

Las obras musicales para ser incluidas en CD ROOM u otros procedimientos multimedia pagarán el valor resultante de aplicar la siguiente fórmula:

$$(PPV) \times (\text{LONGITUD DE MÚSICA}) \times (\text{UNIDADES PRODUCIDAS}) \times \%$$

LONGITUD TOTAL
PPV: precio de venta al público
LONGITUD DE MÚSICA: expresada en mega bytes
LONGITUD TOTAL: longitud total de los archivos del software expresada en mega bytes

%: se determina en función de la importancia de la música: 10% cuando tiene un rol substancial y 6% cuando desempeña un papel complementario.

**DESCRIPCIÓN PÁGINA:
REPRODUCCION**

El diseño de esta página está en html, contiene las tarifas que por este concepto cobra SAYCE.

sayce-6-ut1.html
UBICACIÓN: Sayce\flash\html

1.8.2 DISTRIBUCION



<p>DESCRIPCIÓN PÁGINA: DISTRIBUCIÓN</p> <p>El diseño de esta página está en html, contiene las tarifas que SAYCE cobra por Distribución y puesta a la venta de material discográfico.</p>	<p>sayce-6-ut2.html</p> <p>UBICACIÓN: Sayce\flash\html</p>
--	--

1.8.3 SINCRONIZACION



TARIFAS POR SINCRONIZACIÓN

DERECHO DE SINCRONIZACION

1.- Por la adaptación, arreglo u otra transformación de obras musicales administradas por SAYCE, con miras a su sincronización en mensajes publicitarios de cualquier género, realizadas para ser difundidas a través de cualquier medio de comunicación al público, se establece la tarifa mediante acuerdo bilateral con los usuarios.

Para la utilización lícita de las obras administradas por SAYCE, será menester un contrato individual de licencia, la misma que no afecta los derechos adquiridos con anterioridad por las empresas de edición a través de contratos que hubieren suscrito con socios de SAYCE, ni tampoco los derechos conexos por la utilización en cualquier forma del fonograma.

2.- Obras musicales creadas para mensajes publicitarios (jingles) las obras creadas exclusivamente para mensajes publicitarios en los medio: radio, cine, televisión, unidades móviles, perifoneo y otros medios conocidos o por conocerse pagarán trimestralmente.

- Hasta 20 segundos USA \$20
- Hasta 30 segundos USA \$30
- Hasta 40 segundos USA \$40
- Hasta 60 segundos USA \$50

<p>DESCRIPCIÓN PÁGINA: SINCRONIZACION</p> <p>El diseño de esta página está en html, contiene las tarifas que por sincronización cobra SAYCE.</p>	<p>sayce-6-ut3.html</p> <p>UBICACIÓN: Sayce\flash\html</p>
---	--

1.8.4 COMUNICACIÓN PÚBLICA



SAYCE ECUADOR Un emblema de honor...
...Por el derecho de autor.

TARIFAS POR COMUNICACIÓN PÚBLICA

COMUNICACIÓN PÚBLICA

Por comunicación pública, se entiende todo acto en virtud del cual una pluralidad de personas, reunidas o no en un mismo lugar y, en el momento en que individualmente decidan, puedan tener acceso a la obra sin previa distribución de ejemplares o cada una de ellas. Para el cobro de estos derechos se establece en el siguiente tarifario:

SECTOR ALOJAMIENTO, ALIMENTOS Y BEBIDAS

Los hoteles, residencias, hoteles, apartamentos, hostales residencias, pensiones, cabañas, refugios, albergues, hosterías, paraderos y afines, por difusión o comunicación pública de obras administradas por SAYCE mediante parlantes, abonarán anualmente el resultado de multiplicar el número total de habitaciones que posean por el valor que corresponda de acuerdo a la categoría del establecimiento.

- De lujo (cinco estrellas), el equivalente a US\$3.
- De primera categoría, el equivalente a US\$2
- De segunda categoría, el equivalente a US\$1
- De tercera y cuarta categoría, el equivalente a US\$ 0,50

Este pago no incluye las presentaciones de artistas en vivo que se cobrarán por cada una y de acuerdo al literal cc.

Los restaurantes por difusión pública de obras administradas por SAYCE mediante parlantes, abonarán anualmente, de acuerdo al siguiente detalle:

- De lujo, el equivalente de US\$ 100,00
- De primera el equivalente de US\$ 80,00
- De segunda, el equivalente de US\$ 60,00
- De tercera y cuarta categoría, el equivalente de US\$ 40,00

Este pago no incluye las presentaciones de artistas en vivo que se cobrarán por cada una y de acuerdo al literal cc.

Fuentes de soda, cevicherías, griles, cafeterías y similares, por difusión pública de obras administradas por SAYCE, pagarán anualmente de acuerdo al siguiente detalle:

- De lujo, el equivalente de US\$ 60,00
- De primera el equivalente de US\$ 40,00
- De segunda, el equivalente de US\$ 30,00
- De tercera y cuarta categoría, el equivalente de US\$ 15,00

Este pago no incluye las presentaciones de artistas en vivo que se cobrarán por cada una y de acuerdo al literal cc.

**DESCRIPCIÓN PÁGINA:
COMUNICACIÓN PÚBLICA**

El diseño de esta página está en html, contiene las tarifas que por Comunicación Pública cobra SAYCE a todos los establecimientos comerciales del país.

sayce-6-ut4.html
UBICACIÓN: Sayce\flash\html

1.9 NOTICIAS



SOCIEDAD DE AUTORES Y COMPOSITORES ECUATORIANOS

Noticias ◀ ▶



Los representantes de las sociedades de Gestión Colectiva de Derechos de Autor que conforman el comité Iberoamericano de la Confederación Internacional de Sociedades de Autores y Compositores, CISAC, se darán cita del 12 al 15 de mayo en la ciudad de Quito con motivo de su reunión anual, en la cual se fortalecen la unión y el apoyo que siempre demuestra la CISAC y sus miembros a la gestión de las sociedades de Ibero América, y un espaldarazo especial a la defensa del Derecho de Autor que adelanta la Sociedad de Autores y Compositores Ecuatorianos, SAYCE, en nuestro país.

Damos la bienvenida a Quito a los representantes del Comité Iberoamericano de la CISAC. SAYCE agradece su apoyo en defensa de los derechos de los autores

2008

E-MAIL

DESCRIPCIÓN PÁGINA: NOTICIAS

El diseño de esta página está en flash, contiene: Noticias principales, y detalle de otras noticias inherentes a la Sociedad.

sayce-7.swf

UBICACIÓN: Sayce\flash\html

1.10 WEBMAIL

SAYCE
Sociedad de Autores y
Compositores Ecuatorianos

: Bienvenido, por favor introduzca sus datos :

Usuario: @sayce.com.ec

Clave:

Idioma: Español

Servidor: sayce.com.ec

Entrar >>

Sayce Informatica 2007 <<<|>>> Powered by UebiMiau!

DESCRIPCIÓN PÁGINA: WEBMAIL

El diseño de esta página esta en php, direcciona a un Software Libre Uebimiau que permite el envío y recepción de correo electrónico.

WEBMAIL

UBICACIÓN: <http://mail.sayce.com.ec>

1.11 BUSQUEDA



DESCRIPCIÓN PÁGINA: BUSQUEDA

El diseño de estas páginas está en flash, contiene links de: Búsqueda de Obras, Autores y Biografías de los socios nacionales de SAYCE.

sayce-11.swf

UBICACIÓN: Sayceflash\html LINKS

Link: BUSQUEDA DE OBRAS

Link: BUSQUEDA DE OBRAS

Link: BUSQUEDA DE BIOGRAFIAS

1.11.1 BUSQUEDA DE OBRAS

Busqueda de Obras		
Búsqueda por título de obra :		
<input type="text"/>	<input type="button" value="Buscar"/>	
1 2 3 4		
ISWC	Tema	Ver
T0474059283	A BAILAR EL CARNAVAL	
T0474043338	A JUGAR CARNAVALITO	
T0474078266	A LA VALENTINA EN EL CARNAVAL	
T0474003605	ACHACHAY QUE CARNAVAL	
T0474041467	ALEGRE CARNAVAL	
T0474051427	ALEGRE CARNAVAL	
T0474071594	ALEGRE CARNAVAL	
T0474072779	ALEGRE CARNAVAL	
T0474031714	ALEGRÍAS DE CARNAVAL	
T0474045607	AMORCITO CARNAVAL	
T0474078277	AMORES DE CARNAVAL	
T0474045618	AMORES DE CARNAVAL	
T0474009761	BAILEMOS CARNAVAL	
T0474053274	BANDOLERO CARNAVAL	
T0474045629	CAMINANTE CARNAVAL	
T0474014135	CARNAVAL	
T0474045652	CARNAVAL 76	
T0474038000	CARNAVAL DE AMBATO	

**DESCRIPCIÓN PÁGINA:
BUSQUEDA DE OBRAS**

La construcción de esta página está en php, aquí se puede ingresar el título de la obra que desea buscar.

**BUSQUEDA: sayce-11obra.php
UBICACIÓN: Sayceflash/html**

1.11.1.1 DETALLE OBRAS

Detalle de la obra

[<<< Buscar otra Obra](#)

Información de la obra:	
ObraId :	7807
Nombre Obra :	Carnavalito alegre
ISWC :	1234
Observaciones :	

IP	Nombre	Profesión
171221	Katherine Goyes	Autor

DESCRIPCIÓN PÁGINA: DETALLE OBRA	sayce-11obrainfo.php
Al hacer clic sobre el título de la obra nos presenta el detalle de la obra, es decir, el Código interno, Código ISWC que es asignado por la CISAC, y quienes son los autores y/o compositores de la obra, además consta de una opción que le permite regresar a la búsqueda de obras.	UBICACIÓN: Sayce\flash\html

1.11.2 BUSQUEDA DE AUTORES

Busqueda de Autores

Busqueda por nombre del Autor :

1 2 3

IP	autor	Ver obras
281865731	ABARCA OLEAS HECTOR HUMBERTO	
90008700	ANDRADE ORTEGA CARLOS ABSALON	
281887227	AQUINO SALINAS GERARDO GABRIEL	
281887325	ARAGUNDI ACOSTA JORGE FABIAN	
281888028	ARELLANO FABARA CLELIO ALFREDO	
281895523	BAQUERIZO GONZABAY RAMON TEMISTOCLES	
281896716	BARRENO COBO SABINO RODRIGO	
281897615	BEDOYA ABIGAIL	
281899119	BERMUDEZ QUIJIJE THOMAS ABILIO	
428003491	BERMUDEZ VELASCO EBER ABILIO	
428003687	BONILLA CANDELEJO SANDRA ELIZABETH	
281905064	BRITO RUBIO RAMIRO FABIAN	
281905848	BUSTAMANTE ABRIL JULIO VICENTE	
281906551	CABEZAS ECHEVERRIA LUIS ALBERTO	
281906649	CABEZAS SAMANIEGO MANUEL MARIA	
281907058	CAICEDO AREVALO OLINDA ELIZABETH	
90013123	CALLE SACOTO ABEL	
281910467	CARPIO ABAD RAFAEL MARIA	

**DESCRIPCIÓN PÁGINA:
BUSQUEDA DE AUTOR**

La construcción de esta página está en php, aquí se puede ingresar el nombre del autor que se desea buscar.

sayce-11autor.php

UBICACIÓN: Saycelflash\html

1.11.2.1 DETALLE AUTOR

Detalle del Autor

[<<< Buscar otro Autor](#)

Información del Autor:	
Item :	910
Nombre Autor :	Katherine Goyes
IP :	171221
Pseudonimo :	Katty

ISWC	Título de Obra	Profesión
1234	Carnavalito alegre	Autor
T0474003898	AUSENCIA	Autor

**DESCRIPCIÓN PÁGINA:
DETALLE AUTOR**

Al hacer clic sobre el nombre del autor nos presenta información de su código interno, pseudónimo, y las obras que pertenecen al autor, así como la participación de él en las obras, es decir si es autor o compositor o ambos.

sayce-11autorinfo.php

UBICACIÓN: Sayce\flash\html

1.11.3 BUSQUEDA DE BIOGRAFIA

Busqueda de Biografias

Búsqueda por nombre del autor :

Item	IPI	Nombre	Ver
1	281865731	ABARCA OLEAS HECTOR HUMBERTO	
77	281888028	ARELLANO FABARA CLELIO ALFREDO	
845	281895523	BAQUERIZO GONZABAY RAMON TEMISTOCLES	
166	281910467	CARPIO ABAD RAFAEL MARIA	

**DESCRIPCIÓN PÁGINA:
BUSQUEDA DE BIOGRAFIAS**

La construcción de esta página está en php, aquí se puede ingresar el nombre del autor a buscar de quien se desea conocer la biografía.

sayce-11bio.php

UBICACIÓN: Sayce\flash\html

1.11.3.1 DETALLE BIOGRAFIA

Detalle de Biografía	
<<< Buscar otra Biografía	
Información del Autor:	
AutItem :	910
IPI :	171221
Nombre :	Katherine Goyes
Seudónimo :	
Biografía:	
Lugar de Nacimiento:	Quito
Fecha de Nacimiento:	1975-07-08
Biografía:	Autora intelectual de varias obras de renombre internacional, como la obra más conocida en el mundo es: carnavalito alegre

DESCRIPCIÓN PÁGINA: DETALLE DE BIOGRAFIA

Al hacer clic sobre el nombre del autor en la página anterior se obtiene información como: lugar y fecha de nacimiento y los datos biográficos del socio.

sayce-11bioinfo.php

UBICACIÓN: Sayce\flash\html

2. ADMINISTRADOR DE CATÁLOGOS DE AUTORES/COMPOSITORES, OBRAS, BIOGRAFÍAS

2.1 INGRESO AL ADMINISTRADOR



Username:

Password:

**DESCRIPCIÓN PÁGINA:
INGRESO AL ADMINISTRADOR**

La construcción de esta página está en php, aquí se debe ingresar nombre del usuario y el password para ingresar a la base de datos de la página web, esto es obras, autores y biografías.

index.php

UBICACIÓN: Sayceladm

2.2 MENU DEL ADMINISTRADOR



Módulo de Administración - S.A.Y.C.E.

- [Administrador de Obras](#)
- [Administrador de Autores](#)
- [Administrador de Biografías](#)

DESCRIPCIÓN PÁGINA: MENU ADMINISTRADOR

La construcción de esta página está en php, aquí se presentan tres opciones que tiene el administrador para crear, modificar o eliminar registros en la base de datos como son: Obras, Autores y Biografías.

menu.php

UBICACIÓN: Sayceladm

2.3 ADMINISTRADOR DE OBRAS

Administrador de Obras

<<< Socios

<<< Modulos

Crear una nueva Obra :

Nueva Obra >>>

Busqueda por nombre de obra :

1 2

ISWC	Tema	Editar	Eliminar
T0474026544	ABISMOS DE AUSENCIA		
T0474056502	AMARGA AUSENCIA		
T0474072508	AMARGA AUSENCIA		
T0474003898	AUSENCIA		
T0474018002	AUSENCIA		
T0474018819	AUSENCIA		
T0474021594	AUSENCIA		
T0474037289	AUSENCIA		
T0474041832	AUSENCIA		
T0474042584	AUSENCIA		
T0474049529	AUSENCIA		
T0474060575	AUSENCIA		

**DESCRIPCIÓN PÁGINA:
ADMINISTRADOR DE OBRAS**

La construcción de esta página está en php, en esta pantalla podemos crear una nueva obra, editar la información de una obra o eliminar una obra.

obradmin.php

UBICACIÓN: Sayce\adm\obra

2.3.1 CREANDO OBRA

Creando una obra

[<<< Socios](#)

[<<< Obras](#)

Ingresar la información para grabar nueva obra:

ObraItem :	Automático
Nombre Obra :	<input type="text"/>
ISWC :	<input type="text"/>
Observaciones :	<input type="text"/>

Creando una obra

[<<< Socios](#)

[<<< Obras](#)

Ingresar la información para grabar nueva obra:

ObraItem :	Automático
Nombre Obra :	CARNAVALITO ALEGRE
ISWC :	T123456799
Observaciones :	<input type="text"/>



DESCRIPCIÓN PÁGINA: CREANDO OBRA	obranew.php
La construcción de esta página está en php, en esta pantalla podemos crear una nueva obra.	UBICACIÓN: Sayce\adm\obra

2.3.2 EDITANDO OBRA

Editando una obra

[<<< Socios](#)

[<<< Obras](#)

Información de la obra:

ObraId :	7776
Nombre Obra :	Carnavalito alegre
ISWC :	1234
Observaciones :	

[Editar Información>>>](#)

 [INCLUIR un socio a esta obra >>>](#)

Item	Socioid	Nombre del Socio	Obra	Ip	Excluir
73	895	Katherine Goyes	7776	Autor	

**DESCRIPCIÓN PÁGINA:
EDITANDO OBRA**

La construcción de esta página está en php, en esta pantalla podemos editar la información de la obra, es decir el código ISWC o el título de la obra.

obranew.php

UBICACIÓN: Sayce\adm\obra

2.3.2.1 AÑADIENDO AUTOR A LA OBRA

Creando una obra
[<<< Socios](#)
[<<< Obras](#)

Información de la obra:	
ObraId :	7776
Nombre Obra :	Carnavalito alegre
ISWC :	1234
Observaciones :	

Buscar autor para incluir:

AutorId	ISWC	Autor	Incluir
895	1234	Katherine Goyes	<input type="checkbox"/>

**DESCRIPCIÓN PÁGINA:
AGREGANDO AUTOR A LA OBRA**

La construcción de esta página está en php, en esta pantalla podemos incluir los autores que hayan participado en la creación de la obra, para escoger los autores hacemos clic en el botón buscar y escoger al autor que sea dueño de la obra.

obrAddSocSelectSoc.php

UBICACIÓN: Sayce\adm\obra

2.3.2.2 DEFINIENDO LA PARTICIPACION DEL AUTOR EN LA OBRA

Creando una obra

[<<< Socios](#)

[<<< Obras](#)

Información de la Obra:	
Item :	<input type="text" value="7776"/>
Tema :	<input type="text" value="Carnavalito alegre"/>
ISWC :	<input type="text" value="1234"/>
Observaciones :	<input type="text"/>

Información del Autor:	
AutorId :	<input type="text" value="895"/>
Nombre de Autor :	<input type="text" value="Katherine Goyes"/>
IP :	<input type="text" value="1234"/>
Nombre Alter :	<input type="text" value="Katty"/>

Seleccione el ROL del socio:	
	<div style="border: 1px solid gray; padding: 2px;"><input type="text" value="Autor"/> ▼ Autor Compositor Autor/Compositor</div>

<p style="text-align: center;">DESCRIPCIÓN PÁGINA: ROL DE AUTOR</p> <p>La construcción de esta página está en php, en esta pantalla podemos escoger la participación del autor en la obra, es decir si es autor, compositor o ambos, luego procedemos a grabar obteniendo que una obra quede bien documentada.</p>	<p style="text-align: center;">obrAddSocSelectRol.php</p> <p style="text-align: center;">UBICACIÓN: Sayce\adm\obra</p>
---	--

2.3.3 EXCLUIR UN SOCIO DE UNA OBRA

Excluir socio de una obra
[<<< Admin. de Obras](#)
[<<< Admin. de Socios](#)

Información del Socio:	
SocioItem :	895
Nombre Socio :	Katherine Goyes
IP	1234
Nombre Alter	Katty

**Excluir socio quiere decir que el socio no tendrá participación en esta obra.
Esta acción NO eliminará el socio, si quiere eliminarlo vaya al 'Administrador de Socios'**

**DESCRIPCIÓN PÁGINA:
EXCLUIR SOCIO DE UNA OBRA**

La construcción de esta página está en php, en esta pantalla podemos eliminar la participación de un socio en una obra.


obrDelSoc.php

UBICACIÓN: Sayce\adm\obra









2.4 ADMINISTRADOR DE AUTORES



Crear un nuevo Autor :

 **Nuevo Autor >>>**

Busqueda por nombre del Autor :

IP	Autor	Editar	Eliminar
281865731	ABARCA OLEAS HECTOR HUMBERTO		
281888028	ARELLANO FABARA CLELIO ALFREDO		
281895523	BAQUERIZO GONZABAY RAMON TEMISTOCLES		
281910467	CARPIO ABAD RAFAEL MARIA		

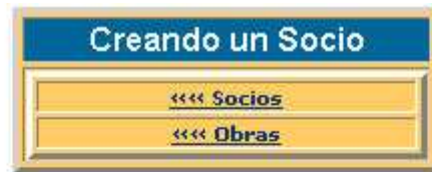
**DESCRIPCIÓN PÁGINA:
ADMINISTRADOR DE AUTORES**

La construcción de esta página está en php, en esta pantalla podemos realizar la creación, búsqueda, modificación de información y eliminación de un autor o socio.

socAdmin.php

UBICACIÓN: Sayce\adm\socio

2.4.1 CREANDO SOCIO



Ingresar la información para grabar nuevo socio:	
SocioItem :	Automático
Nombre Socio :	Katherine Goye
IP de Socio :	1234
Nombre alternativo :	Katty
<input type="button" value="Crear Nuevo Socio >>>"/>	



DESCRIPCIÓN PÁGINA: CREANDO NUEVO SOCIO

La construcción de esta página está en php, en esta pantalla se ingresan los datos del socio como son Nombre, apellido, Código IP y pseudónimo, luego procedemos a presionar crear socio nuevo para grabar los datos ingresados.

socNew.php

UBICACIÓN: Sayce\adm\socio

2.4.2 EDITANDO SOCIO

Editando Autor

[<<< Autores](#)
[<<< Obras](#)

Información del Autor:

Item :	894	
Nombre Autor :	Katherine Goyes	
IP :	1234	
Nombre alter :	Katty	

[INCLUIR una obra a este Autor >>>](#)

DetItem	Obraltem	Nombre de Obra	SocioItem	TipoIP	Excluir
71	6	AUSENCIA	894	Autor	

Editando un Socio

[<<< Socios](#)
[<<< Obras](#)

Editar ahora la información del socio:

SocioItem :	894	
Nombre Socio :	Katherine Goyes	
ISWC :	1234	
Observaciones :	Katty	

DESCRIPCIÓN PÁGINA: EDITANDO SOCIO

La construcción de esta página está en php, en esta pantalla se pueden editar los datos del socio, así como incluir o excluir obras en las que tenga participación el socio.

socEdit.php

UBICACIÓN: Sayceladm\socio

2.4.3 AGREGANDO OBRA AL SOCIO



Información del Autor:	
Item :	894
Nombre Socio :	Katherine Goyes
IP :	1234
Nombre Alter :	Katty

Buscar obra para incluir:			
<input type="text"/>	Buscar		
1 2			
Obraltem	ISWC	Nombre Obra	Incluir
2805	T0474026544	ABISMOS DE AUSENCIA	
5923	T0474056502	AMARGA AUSENCIA	
7576	T0474072508	AMARGA AUSENCIA	
6	T0474003898	AUSENCIA	
1909	T0474018002	AUSENCIA	
1990	T0474018819	AUSENCIA	Incluir obra a este socio
2285	T0474021594	AUSENCIA	

DESCRIPCIÓN PÁGINA: AGREGANDO OBRA AL SOCIO

La construcción de esta página está en php, Luego de presionar el botón incluir obra de la página anterior, procedemos a buscar la obra que queremos incluir al socio y presionamos la opción incluir.

socAddObrSelectObr.php

UBICACIÓN: Sayce\adm\socio

2.4.3 EXCLUIR OBRA DE UN SOCIO

Editando un Socio

[<<< Socios](#)

[<<< Obras](#)

Información de la Obra:	
ObraItem :	6
Nombre Obra :	AUSENCIA
ISW :	T0474003898
Nombre Alter :	

Excluir obra quiere decir que la obra no tendrá participación con el socio.
Esta acción NO eliminará la Obra, si quiere eliminarlo vaya al 'Administrador de Obras'

Excluir Obra >>>

DESCRIPCIÓN PÁGINA: EXCLUYENDO OBRA DEL SOCIO

La construcción de esta página está en php, Luego de presionar el botón excluir de la página Editando Socio, procedemos a buscar la obra que queremos excluir al socio.

socDelObr.php

UBICACIÓN: Sayce\adm\socio

2.4.4 ELIMINANDO UN SOCIO

Eliminando un Socio

[<<< Socios](#)

[<<< Obras](#)

Información del socio:

Item :	894
Nombre Autor :	Katherine Goyes
IP :	1234
Nombre Alter :	Katty

DetItem	Obraltem	Nombre de la Obra	SocioItem	TipoIp
71	6	AUSENCIA	894	Autor

Eliminará un socio
Esta acción **NO** eliminará las obras, si quiere eliminarlas vaya al 'Administrador de Obras'

Eliminar Socio >>

DESCRIPCIÓN PÁGINA: ELIMINANDO SOCIO

La construcción de esta página está en php, Luego de presionar el botón eliminar al ingresar al administrador de autores, obtenemos la pantalla anterior en la cual al presionar eliminar socio, habremos borrado de la base de datos el nombre del socio y su datos, más no sus obras, ya que estas pueden estar enlazadas a otros socios.

socDelete.php

UBICACIÓN: Sayceladm\socio

2.5 ADMINISTRADOR DE BIOGRAFIAS

Administrador de Biografías

<<< [Modulos](#)

Busqueda por nombre del autor :

Item	IPI	Nombre	Ver
910	171221	Katherine Goyes	

**DESCRIPCIÓN PÁGINA:
ADMINISTRADOR DE BIOGRAFIAS**

La construcción de esta página está en php, Esta pantalla nos permite ingresar el nombre del socio del cual queremos obtener la biografía.

bioAdmin.php

UBICACIÓN: Sayceladm\bio

2.5.1 DETALLE DE BIOGRAFIA



Información del Autor:	
AutItem :	910
IPI :	171221
Nombre :	Katherine Goyes
Seudónimo :	

Biografía:	
Lugar de Nacimiento:	Quito
Fecha de Nacimiento:	1975-07-08
Biografía:	Autora intelectual de varias obras de renombre internacional, como la obra más conocida en el mundo es: carnavalito alegre



DESCRIPCIÓN PÁGINA: DETALLE DE BIOGRAFIA

La construcción de esta página está en php, Al presionar el botón de la pantalla anterior obtenemos la pantalla del detalle de la biografía, donde constan los datos del socio, así como su biografía.

bioView.php

UBICACIÓN: Sayce\adm\bio

2.5.2 EDITANDO BIOGRAFIA

Editando Biografías	
<<< Biografías	
Información del Autor:	
AutItem :	910
IPI :	171221
Nombre :	Katherine Goyes
Seudónimo :	
Biografía:	
Lugar de Nacimiento:	Quito-Ecuador
Fecha de Nacimiento:	1975-08-07
Biografía:	Autora intelectual de varias obras de renombre internacional, como la obra más conocida en el mundo es: carnavalito alegre
 Editar Biografía >>>	
 Eliminar Biografía >>>	

**DESCRIPCIÓN PÁGINA:
EDITANDO BIOGRAFIA**

La construcción de esta página está en php, En esta pantalla tenemos los botones editar y eliminar biografía.

bioView.php

UBICACIÓN: Sayceladm\bio

2.5.2.1 EDITANDO BIOGRAFIA

Editando Biografía

«««« **Biografías**

Información del Autor:	
AutItem :	910
IPI :	171221
Nombre :	Katherine Goyes
Seudónimo :	

Editar ahora la Biografía:	
Lugar de Nacimiento:	Quito-Ecuador
Fecha de Nacimiento (aaaa-mm-dd):	1975-08-07
Biografía:	<p>Autora intelectual de varias obras de renombre internacional, como la obra más conocida en el mundo es: carnavalito alegre</p>

Guardar Biografía >>>

<p>DESCRIPCIÓN PÁGINA: EDITANDO BIOGRAFIA</p> <p>La construcción de esta página está en php, Al presionar el botón Editar Biografía podemos modificar la información del autor, como su biografía y procedemos a Guardar Biografía.</p>	<p style="text-align: center;">bioView.php</p> <p style="text-align: center;">UBICACIÓN: Sayce\adm\bio</p>
--	--

2.5.3 CREANDO NUEVA BIOGRAFIA

Editando Biografías

[<<< Biografías](#)

Información del Autor:	
AutItem :	<input type="text" value="910"/>
IPI :	<input type="text" value="171221"/>
Nombre :	<input type="text" value="Katherine Goyes"/>
Seudónimo :	<input type="text"/>

[NO EXISTE BIOGRAFÍA, clic aquí para crearla >>>](#)

<p style="text-align: center;">DESCRIPCIÓN PÁGINA: CREANDO NUEVA BIOGRAFIA</p> <p>La construcción de esta página está en php, Al presionar el botón ver del Administrador de Biografías podemos obtener también la pantalla de que no existe biografía y nos da la opción de crear una nueva biografía.</p>	<p>bioNew.php</p> <p>UBICACIÓN: Sayce\adm\bio</p>
--	---

2.5.4. ELIMINANDO BIOGRAFÍAS

Editando Biografías

[<<< Biografías](#)

Información del Autor:	
AutItem :	910
IPI :	171221
Nombre :	Katherine Goyes
Seudónimo :	

Eliminará la siguiente Biografía:

Lugar de Nacimiento:	Quito-Ecuador
Fecha de Nacimiento:	1975-08-07
Biografía:	Autora intelectual de varias obras de renombre internacional, como la obra más conocida en el mundo es: carnavalito alegre

Eliminará la Biografía
Esta acción NO eliminará el socio, si quiere eliminarlo vaya al 'Administrador de Socios'

Eliminar la Biografía >>>

**DESCRIPCIÓN PÁGINA:
ELIMINANDO BIOGRAFIA**

La construcción de esta página está en php, Al presionar el botón eliminar Biografía, obtenemos otra pantalla en la cual nos pide confirmación de dicha eliminación.

bioDelete.php

UBICACIÓN: Sayce\adm\bio

ANEXO B

Manual Técnico

CONTENIDO DEL MANUAL TÉCNICO

ANÁLISIS Y DISEÑO USANDO METODOLOGÍA ADPD CON UML - OMG 144

El proyecto en resumen:	145
MODELO DEL NEGOCIO - KITE LEVEL	146
Parte I: Diagrama de Casos De Uso del Negocio – Kite Level	146
Parte II: Especificación textual de Casos de Uso del Negocio-Kite Level	147
1 Buscar en el Catálogo de Autor/Compositor, Obra, Biografía	147
2 Navegar el Contenido Web de la SAYCE	147
3 Requerir afiliación	147
4 Tramitar la afiliación de Autor/Compositor	147
5 Administrar la afiliación del Autor/Compositor	147
6 Administrar el Catálogo de Autor/Compositor,	147
7 Administrar el Contenido Web de la SAYCE	148
8 Administrar Servidor LAMP	148
Parte III: Diagramas de Actividad del Negocio – Kite Level	149
1 Buscar en el Catálogo de Autor/Compositor, Obra, Biografía	149
2 Navegar el Contenido Web de la SAYCE	149
3 Requerir afiliación	150
4 Tramitar la afiliación de Autor/Compositor	150
5 Administrar la afiliación del Autor/Compositor	151
6 Administrar el Catálogo de Autor/Compositor, Obra, Biografía	151
7 Administrar el Contenido Web de la SAYCE	152
8 Administrar Servidor LAMP	152
MODELO DEL SISTEMA – SEA LEVEL	153
Parte IV: Diagrama de casos de uso del Sistema - Sea Level	153
Parte V: Especificación textual de Casos de Uso del Sistema–Sea Level	154
1 Buscar en el Catálogo de Autor/Compositor, Obra, Biografía	154
1.1 Buscar Autor/Compositor	154
1.2 Buscar Obra	154
1.3 Buscar Biografía	154
6 Administrar el Catálogo de Autor/Compositor, Obra, Biografía	155
6.1 Validar Usuario y Password	155
6.2 Administrar Autor/Compositor	155
6.3 Administrar Obra	155
6.4 Administrar Biografía	156
6.5 Administrar Operadores	156
Parte VI: Diagramas de Actividad del Sistema – Sea Level	157
1 Buscar en el Catálogo de Autor/Compositor, Obra, Biografía	157
1.1 Buscar Autor/compositor	157
1.2 Buscar Obra	158
1.3 Buscar Biografía	158
6 Administrar el Catálogo de Autor/Compositor, Obra, Biografía	159
6.1 Validar Usuario y Password	159
6.2 Administrar Autor/Compositor	160
6.3 Administrar Obra	160
6.4 Administrar Biografía	161
6.5 Administrar Operadores	161
MODELO DEL SISTEMA – FISH LEVEL	162
Parte VII: Diagrama de casos de uso del Sistema - Fish Level	162
Parte VIII: Diagrama de clases del Sistema – Fish Level	165
Parte IX: Diagrama de implantación – Fish Level	166
Parte X: Diagrama de implantación – Fish Level	167

ANÁLISIS Y DISEÑO USANDO METODOLOGÍA ADPD CON UML - OMG

La metodología ADPD quiere decir Desarrollo de Proyecto ÁGIL y Definido, ésta es fruto del aprovechamiento de las metodología clásicas y las más populares metodologías Ágiles que hoy en día parecen ser: Extreme Programming (XP), Scrum, Feature Driven Development (FDD), Lean Software Development, Agile Unified Process (Agile UP or AUP), Crystal, and Dynamic Systems Development Method (DSDM).

El desarrollo de software ÁGIL es un estilo de desarrollo caracterizado por poner énfasis en la gente, la comunicación, el software trabajado, y la respuesta a los cambios. Todas las metodologías ÁGILES involucran un flujo de trabajo iterativo y entrega incremental del software trabajado en grupos de tiempos pequeños iterativos. Una iteración es esencialmente una corta liberación del software. Generalmente durante cada iteración algunas actividades ocurrirán en paralelo, tales como requerimientos, codificación, testeo. Las iteraciones son típicamente de alcance fijo (aunque este alcance varía entre las metodologías) y así son referidas como grupo de tiempo. El tiempo asignado para cada iteración es algunas veces referido como ciclo de tiempo.

Las metodologías ÁGILES también difieren en las actividades y artefactos que estos producen. Las metodologías que tienen muchos artefactos (o productos del trabajo realizado) y pasos a lo largo de manera que son referidos como alta ceremonia. Lo contrario, las metodologías con menos énfasis en aquello son referidos como de baja ceremonia. El número de gente involucrado también ayuda a definir la cantidad de ceremonia al rededor de una particular metodología. Es importante atinar un apropiado balance en cualquiera sea en contexto.

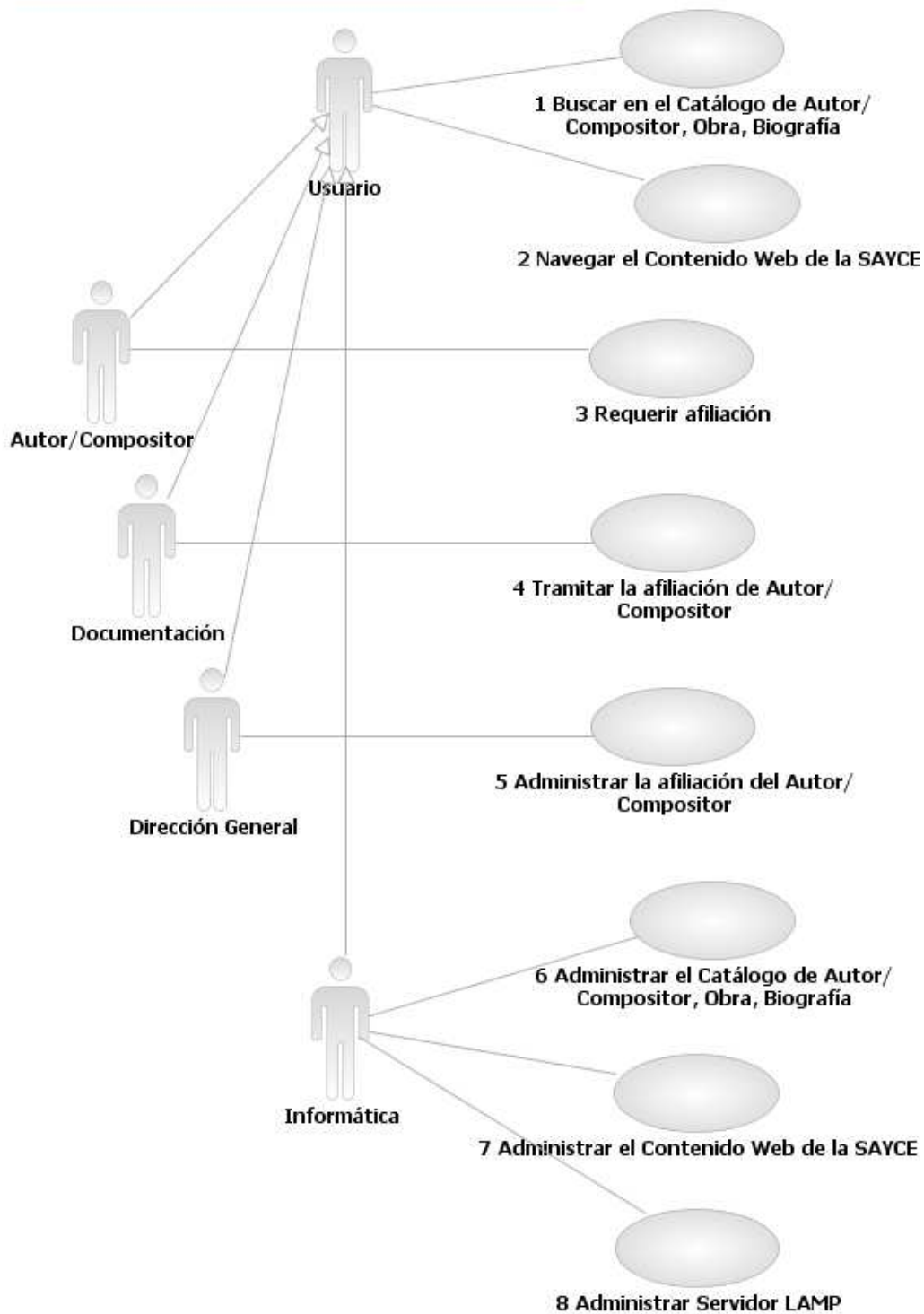
El proyecto en resumen:

Paradigma: Orientado a Objetos Ciclo De Vida: ADPD		Vistas OMG Metodología: ADPD-OMG	Diagramas UML2.0 Método: ADPD-UML2.0	Herramientas	Prácticas: ADPD
Fase Walking Skeleton (WS)	Análisis	Modelo del negocio	✓ Casos de Uso de Negocio ✓ Casos de Uso Textual	Rational Systems Developers	<ul style="list-style-type: none"> ○ Pequeñas liberaciones ○ Cliente en el sitio ○ Planificación del evento ○ Gerencia de requerimientos
	Diseño	Modelo de estructura	✓ Clases ✓ Componentes ✓ Paquetes ✓ Implantación		
	Desarrollo	Modelo 3 capas (sólo estructura)	Presentación	HTML (embebido PHP)	<ul style="list-style-type: none"> ○ Programación en par ○ Estándares de codificación ○ Sistema Versión Concurrente
			Dominio (negocio)	PHP	
			Acceso a Datos	MySQL	
	Test	Qa Quality Assurance	Pruebas	Servidor LAMP (Linux, Apache, MySQL, PHP)	<ul style="list-style-type: none"> ○ Test Unitario ○ Test de Suite ○ Test Funcional
Implantación/ Liberación			Servidor LAMP	<ul style="list-style-type: none"> ○ Testeo de Regresión 	
Fase de Liberación 1	Análisis	Modelo del negocio	✓ Casos de Uso de Negocio ✓ Casos de Uso Textual	Rational Systems Developers	<ul style="list-style-type: none"> ○ Pequeñas liberaciones ○ Cliente en el sitio ○ Planificación del evento ○ Gerencia de requerimientos
	Diseño	Modelo de estructura	✓ Clases ✓ Componentes ✓ Paquetes ✓ Implantación		
		Modelo de comportamiento	✓ Casos de Uso de Sistema ✓ Actividades		
	Desarrollo	Modelo 3 capas (Funcionalidad1)	Presentación	HTML (embebido PHP)	<ul style="list-style-type: none"> ○ Programación en par ○ Estándares de codificación ○ Sistema de Versión Concurrente
			Dominio (negocio)	PHP	
			Acceso a Datos	MySQL	
Test	Qa Quality Assurance	Pruebas	Servidor LAMP	<ul style="list-style-type: none"> ○ Test Unitario ○ Test de Suite ○ Test Funcional 	
Implantación/ Liberación			Servidor LAMP	<ul style="list-style-type: none"> ○ Testeo de Regresión 	
Fase de Liberación 2 * Funcionalidad 2					
⋮					
Fase de Liberación n-1 * Funcionalidad n-1					
Fase de Liberación n * Proyecto concluido					

MODELO DEL NEGOCIO - KITE LEVEL

Parte I: Diagrama de Casos De Uso del Negocio – Kite Level

Parte I: Diagrama de Casos De Uso del Negocio - Kite Level



Parte II: Especificación textual de Casos de Uso del Negocio-Kite Level

1 Buscar en el Catálogo de Autor/Compositor, Obra, Biografía

Kite level

1. El usuario abre el portal Web de la SAYCE en pantalla.
2. El servidor responde petición.
3. El usuario visualiza opciones de catálogos.

2 Navegar el Contenido Web de la SAYCE

Kite level

1. El usuario solicita una página desde el Contenido Web de la SAYCE.
2. El servidor Web tramita la página Web solicitada por del usuario
3. El servidor Web recibe en el navegador la página solicitada

3 Requerir afiliación

Kite level

1. El Autor/Compositor llena datos de solicitud de afiliación
2. El Autor/Compositor envía una solicitud de afiliación a Documentación.
3. Documentación *tramita la solicitud de afiliación de Autor/Compositor (1.2.4).*
4. El Autor/Compositor recibe el estado de la solicitud de afiliación

4 Tramitar la afiliación de Autor/Compositor

Kite level

1. El Autor/Compositor *requiere afiliación* a Documentación.
2. Documentación recibe la solicitud de afiliación del Autor/Compositor.
3. Documentación envía la solicitud de afiliación a la Dirección General.
4. Dirección General *administra la afiliación del Autor/Compositor.*
5. Documentación recibe el estado de afiliación desde la Dirección General.
- 5.1.1 Respuesta positiva: Documentación envía los datos necesarios para registrar en el Catálogo a informática.
- 5.1.2 Informática *administra el catálogo de Autor/Compositor, obra, biografía*
6. Documentación envía el estado de afiliación al Autor/Compositor.

5 Administrar la afiliación del Autor/Compositor

Kite level

1. Documentación *tramita la afiliación del Autor/Compositor* en la Dirección General.
2. Dirección General recibe la solicitud de afiliación desde Documentación.
3. Dirección General estudia y ejecuta la solicitud de afiliación.
4. Dirección General envía notificación del estado de la afiliación a Documentación

6 Administrar el Catálogo de Autor/Compositor,

Kite level

1. Informática abre el administrador de Catálogos de la SAYCE en pantalla.
2. El servidor responde petición.
- 3 Informática visualiza pantalla de acceso.

7 Administrar el Contenido Web de la SAYCE

Kite level

Descripción:

1. Informática recepta peticiones de modificación del Contenido Web de la SAYCE desde sus diferentes departamentos.
2. Informática hace el Análisis, diseño, desarrollo, test, implantación/liberación de la modificación del Contenido Web de la SAYCE.
3. Informática aplica cambios en el servidor LAMP
4. El servidor publica los cambios

8 Administrar Servidor LAMP

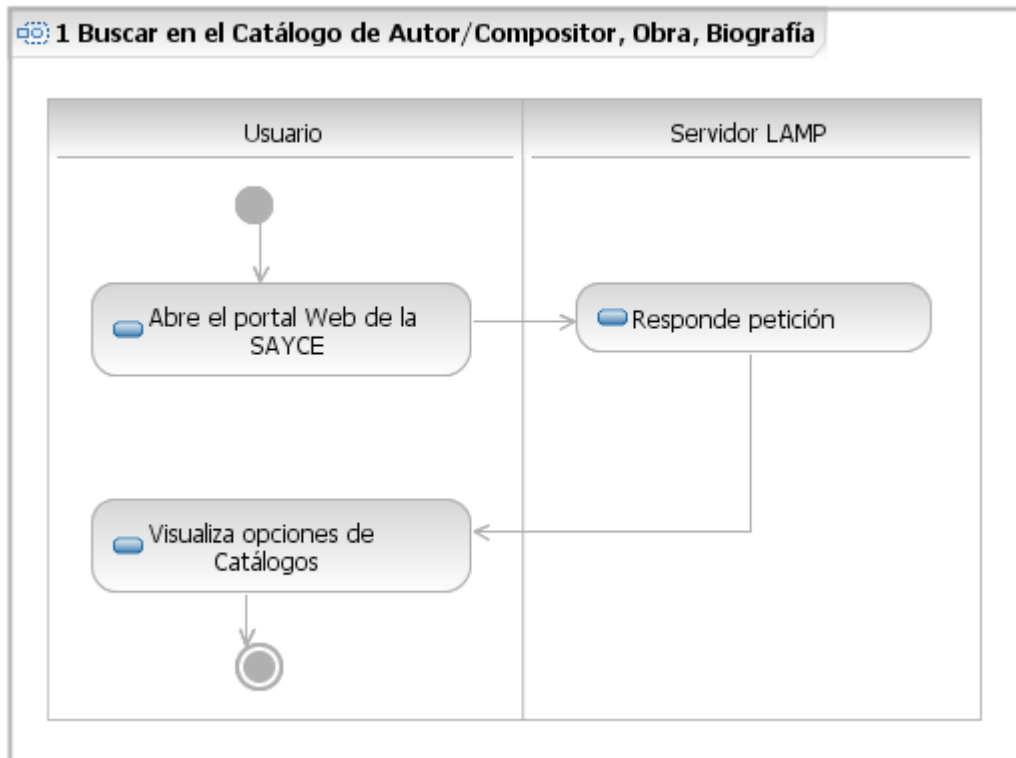
Kite level

Descripción:

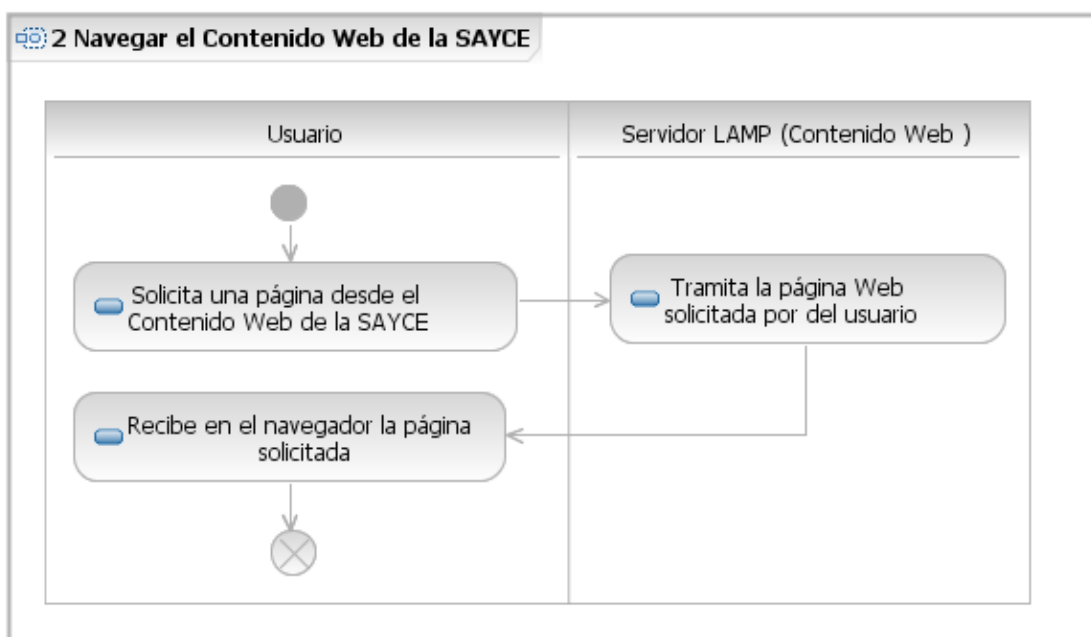
1. Informática realiza actualizaciones a la estructura del servidor LAMP (Linux, Apache, MySql, PHP)
2. El servidor aplica actualizaciones

Parte III: Diagramas de Actividad del Negocio – Kite Level

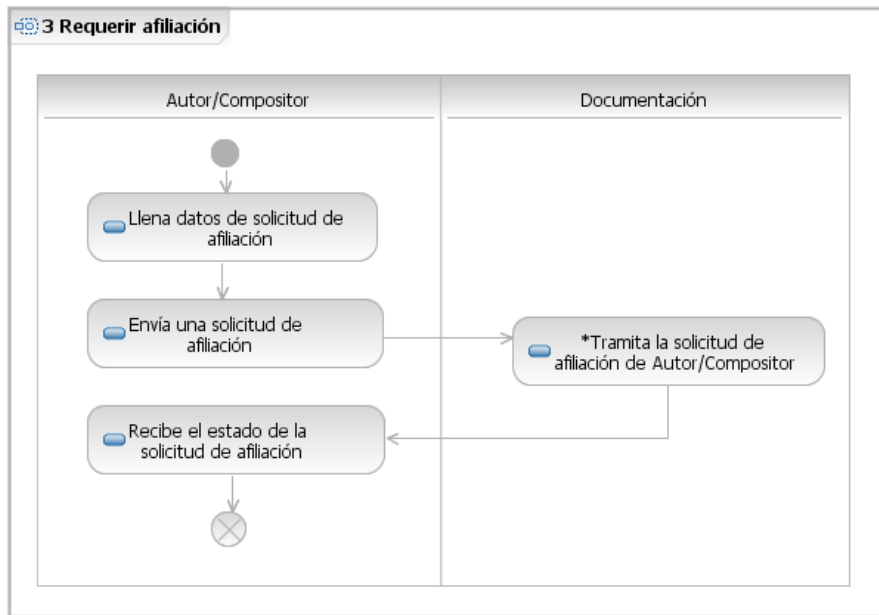
1 Buscar en el Catálogo de Autor/Compositor, Obra, Biografía



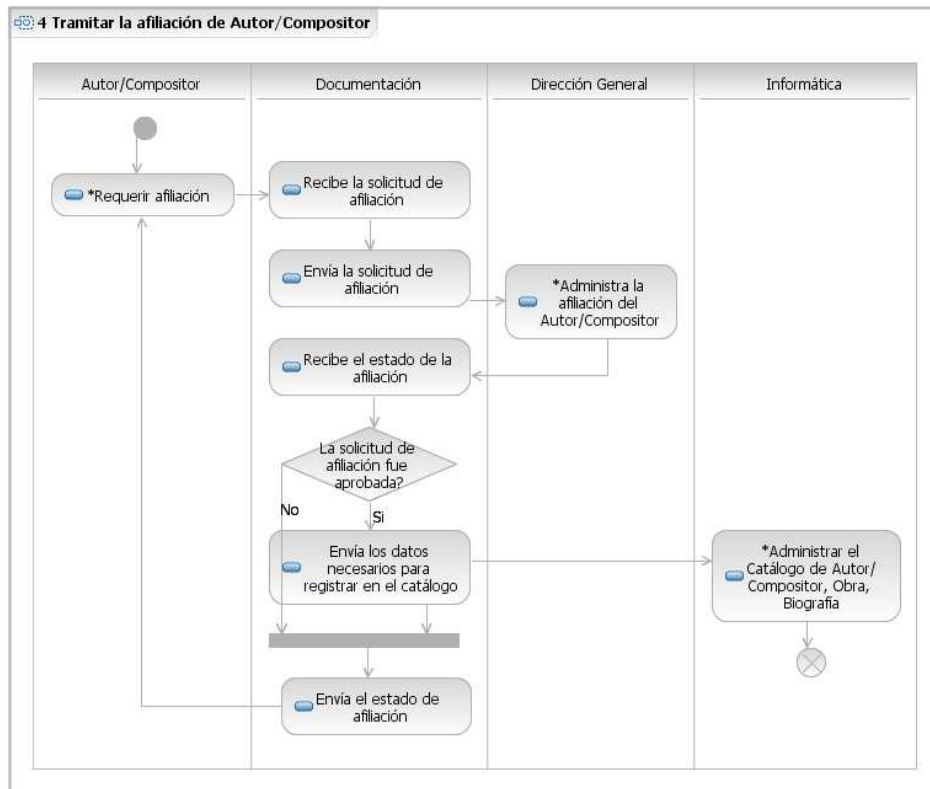
2 Navegar el Contenido Web de la SAYCE



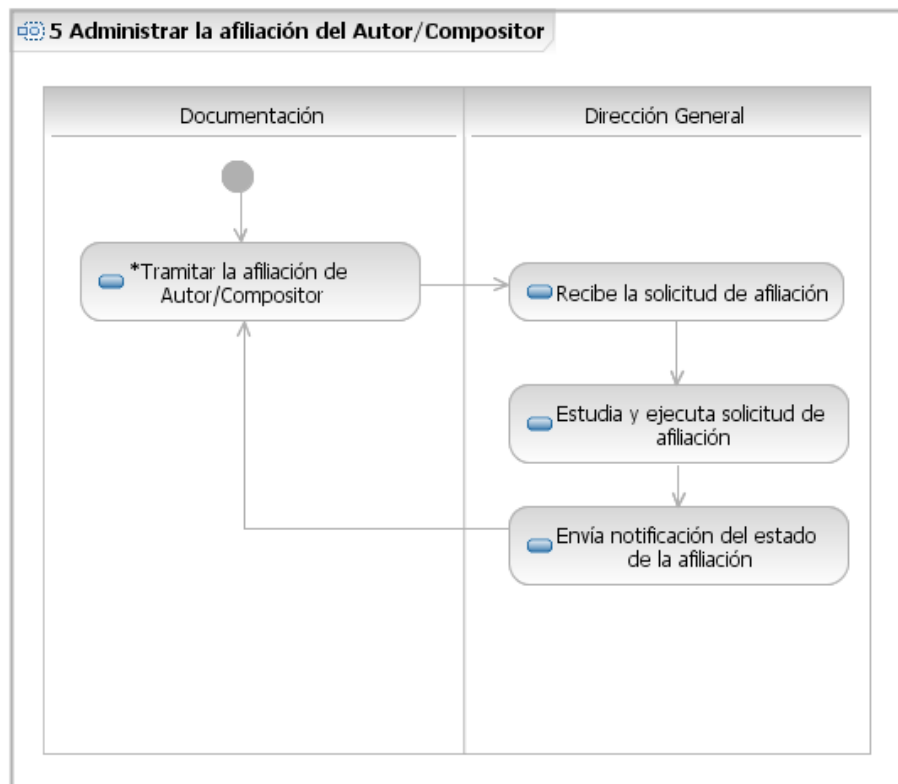
3 Requerir afiliación



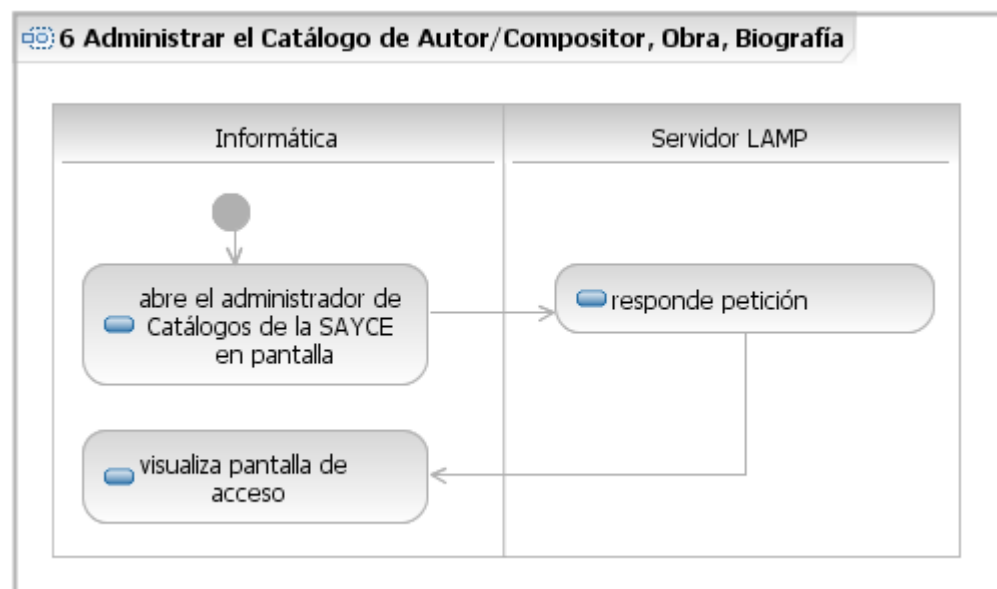
4 Tramitar la afiliación de Autor/Compositor



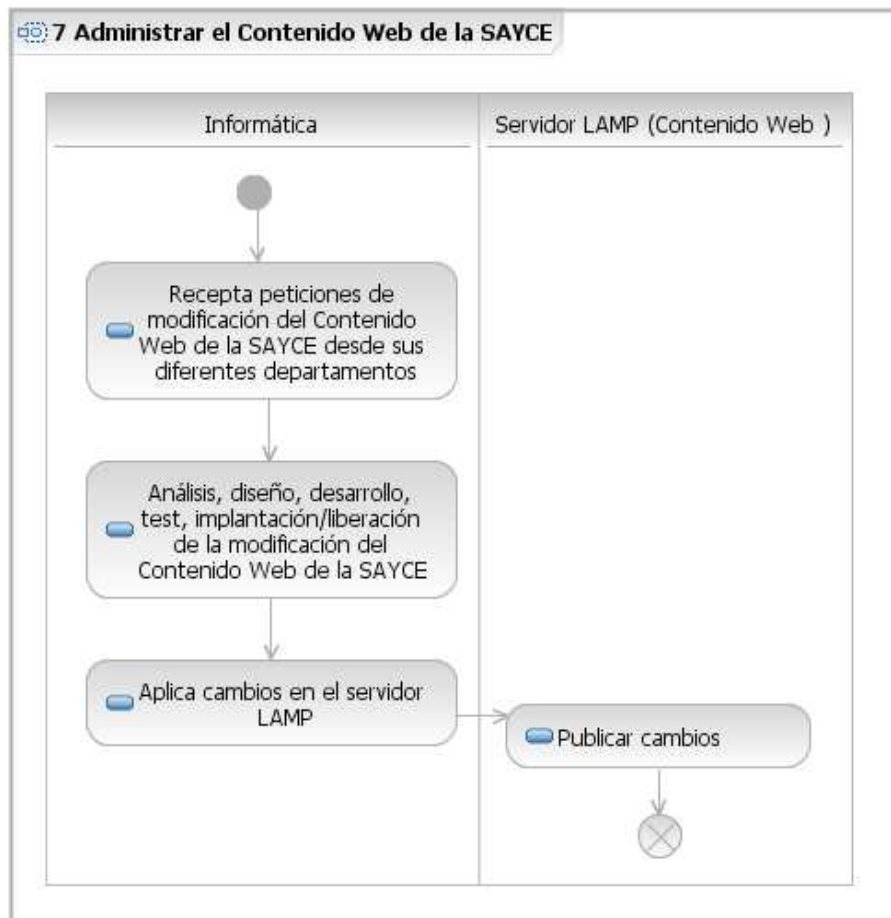
5 Administrar la afiliación del Autor/Compositor



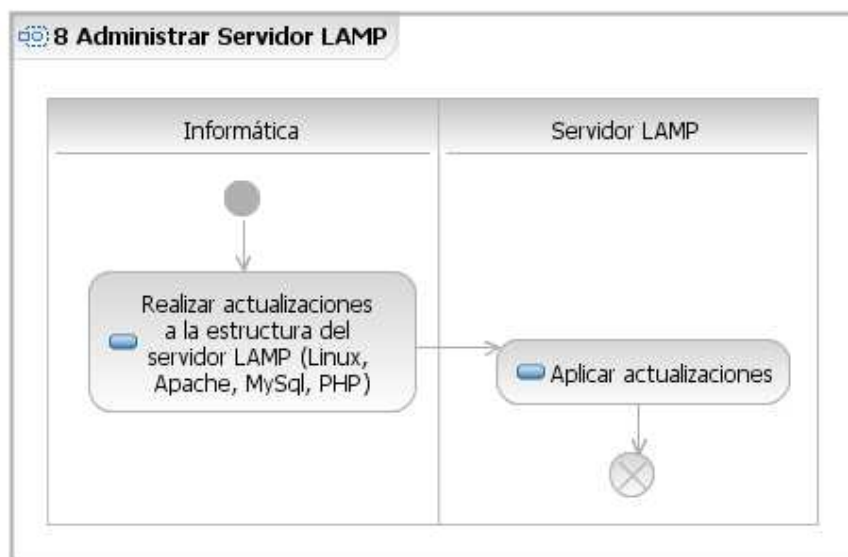
6 Administrar el Catálogo de Autor/Compositor, Obra, Biografía



7 Administrar el Contenido Web de la SAYCE



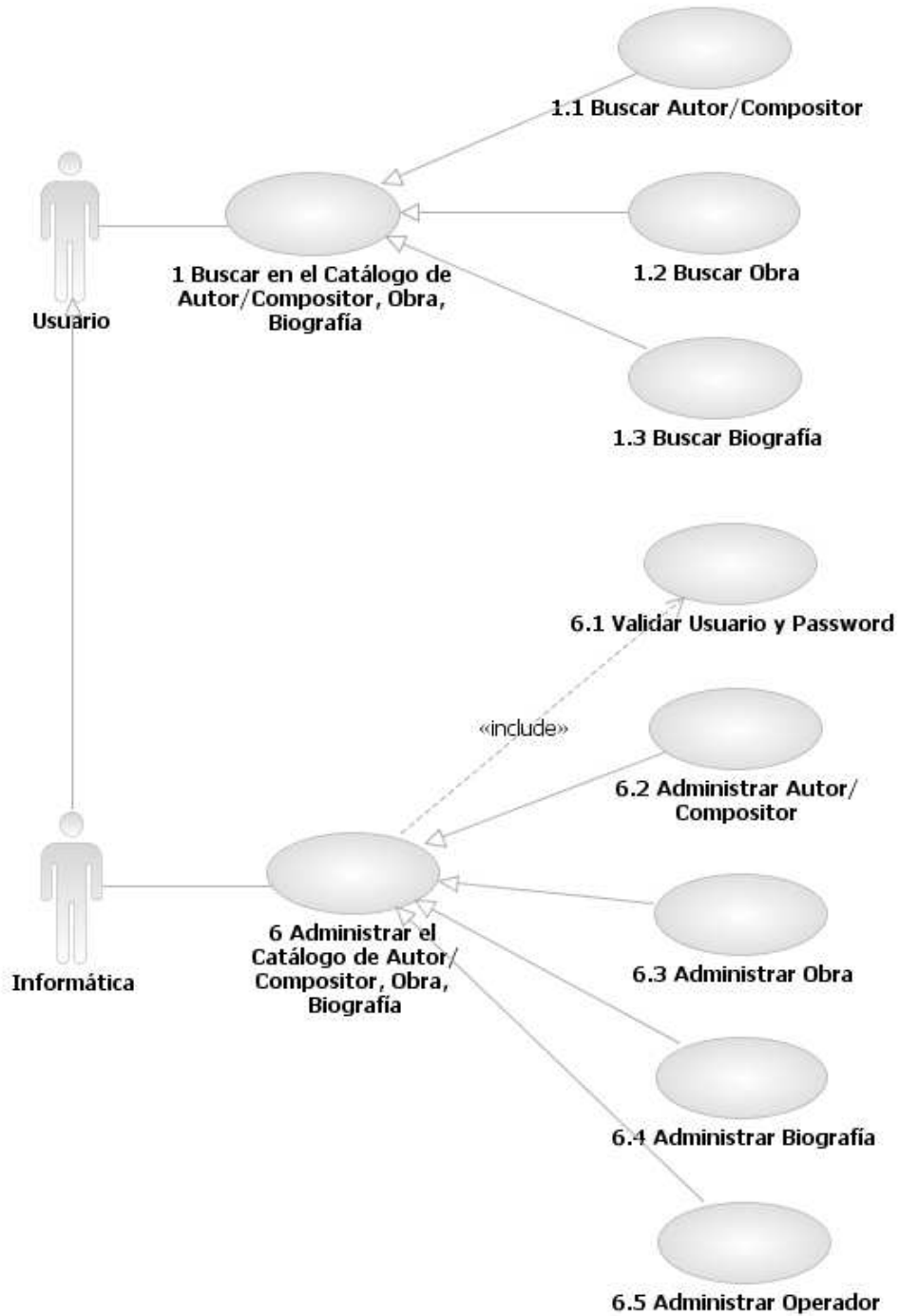
8 Administrar Servidor LAMP



MODELO DEL SISTEMA – SEA LEVEL

Parte IV: Diagrama de casos de uso del Sistema - Sea Level

Parte IV: Diagrama de Casos De Uso Del Sistema - Sea Level



Parte V: Especificación textual de Casos de Uso del Sistema–Sea Level

1 Buscar en el Catálogo de Autor/Compositor, Obra, Biografía

Kite level

1. El usuario abre el portal Web de la SAYCE en pantalla.
2. El servidor responde petición.
3. El usuario visualiza opciones de catálogos.

Generalizaciones:

- 1.1 *Buscar Autor/Compositor*
- 1.2 *Buscar Obra*
- 1.3 *Buscar Biografías*

1.1 Buscar Autor/Compositor

Sea level

1. El usuario busca el nombre de un Autor/Compositor para informarse.
2. El servidor verifica la existencia de Autor/Compositor.
- 3 *No existe* Autor/Compositor:
 - 3.1 El usuario visualiza mensaje de error.
- 3 *Si existe* Autor/Compositor:
 - 3.2 El usuario visualiza un listado de Autores/Compositores en la pantalla.
 - 3.3 El usuario selecciona ver Autor/Compositor.
 - 3.4 El servidor responde petición.
 - 3.5 El usuario visualiza Autor/Compositor en la pantalla.

1.2 Buscar Obra

Sea level

1. El usuario busca el nombre de una Obra para informarse.
2. El servidor verifica la existencia de Obra.
- 3 *No existe* Obra:
 - 3.1 El usuario visualiza mensaje de error.
- 3 *Si existe* Obra:
 - 3.2 El usuario visualiza un listado de Obras en la pantalla.
 - 3.3 El usuario selecciona ver Obra.
 - 3.4 El servidor responde petición.
 - 3.5 El usuario visualiza Obra en la pantalla.

1.3 Buscar Biografía

Sea level

- 1 El usuario busca el nombre de Autor/Compositor para informarse de su Biografía.
- 2 El servidor verifica la existencia de Autor/Compositor.
- 3 *No existe* Autor/Compositor:
 - 3.1 El usuario visualiza mensaje de error.
- 3 *Si existe* Autor/Compositor:
 - 3.2 El usuario visualiza listado de Autores/Compositor en pantalla
 - 3.3 El usuario selecciona ver Biografía.
 - 3.4 El servidor verifica la existencia de Biografía.
 - 3.5: *No existe* Biografía:
 - 3.5.1 El usuario visualiza mensaje de error.
 - 3.5: *Si existe* Biografía:
 - 3.5.2 El usuario visualiza Biografía del Autor/Compositor en pantalla.

6 Administrar el Catálogo de Autor/Compositor, Obra, Biografía

Kite level

1. Informática abre el administrador de Catálogos de la SAYCE en pantalla.
2. El servidor responde petición.
- 3 Informática visualiza pantalla de acceso.
- 4 Incluye 6.1 *Validar Usuario y Password*

Generalizaciones:

- 6.2 *Administrar Autor/Compositor.*
- 6.3 *Administrar Obra.*
- 6.4 *Administrar Biografía.*
- 6.5 *Administrar Operador*

6.1 Validar Usuario y Password

Sea level

- 1 Informática ingresa el Usuario y Password para acceder al sistema.
- 2 El servidor verifica existencia de Usuario y Password.
- 3 *No existe* Usuario y Password:
 - 3.1 Informática visualiza mensaje de error.
- 3 *Si existe* Usuario y Password:
 - 3.2 Informática visualiza el administrador de catálogos

6.2 Administrar Autor/Compositor

Sea level

- 1 Informática busca el nombre del Autor/Compositor para administrar.
- 2 El servidor verifica la existencia de Autor/Compositor.
- 3 *No existe* Autor/Compositor:
 - 3.1 Informática visualiza mensaje de error.
- 3 *No existe* Autor/Compositor:
 - 3.2 Informática selecciona agregar Autor/Compositor.
- 3 *Si existe* Autor/Compositor:
 - 3.3 Informática selecciona editar Autor/Compositor.
- 3 *Si existe* Autor/Compositor:
 - 3.4 Informática selecciona eliminar Autor/Compositor.

6.3 Administrar Obra

Sea level

1. Informática busca el nombre de la Obra para administrar.
2. El servidor verifica la existencia de Obra.
- 3 *No existe* Obra:
 - 3.1 Informática visualiza mensaje de error
- 3 *No existe* Obra:
 - 3.2 Informática selecciona agregar Obra.
- 3 *Si existe* Obra:
 - 3.3 Informática selecciona editar Obra.
- 3 *Si existe* Obra:
 - 3.4 Informática selecciona eliminar Obra.

6.4 Administrar Biografía

Sea level

1. Informática busca el nombre del Autor/Compositor para administrar su Biografía.
2. El servidor verifica la existencia de Autor/Compositor.
- 3 *No existe* Autor/Compositor.
 - 3.1 Informática recibe mensaje de error.
- 3 *Si existe* Autor/Compositor.
 - 3.2 Informática visualiza listado de Autor/Compositor.
 - 3.3 Informática selecciona ver Biografía.
 - 3.4. El servidor verifica la existencia de la Biografía.
 - 3.5 *No existe* Biografía:
 - 3.5.1 Informática visualiza mensaje de error.
 - 3.5.2 Informática selecciona agregar Biografía.
 - 3.5 *Si existe* Biografía:
 - 3.5.2 Informática selecciona editar Biografía.
 - 3.5 *Si existe* Biografía:
 - 3.5.3 Informática selecciona eliminar Biografía.

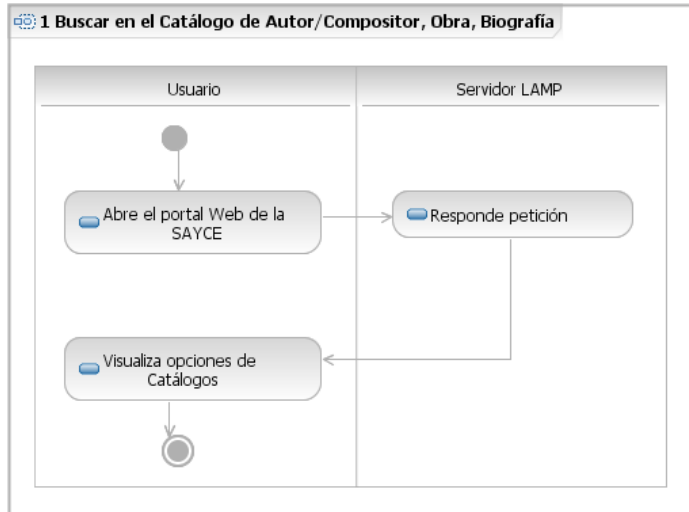
6.5 Administrar Operadores

Sea level

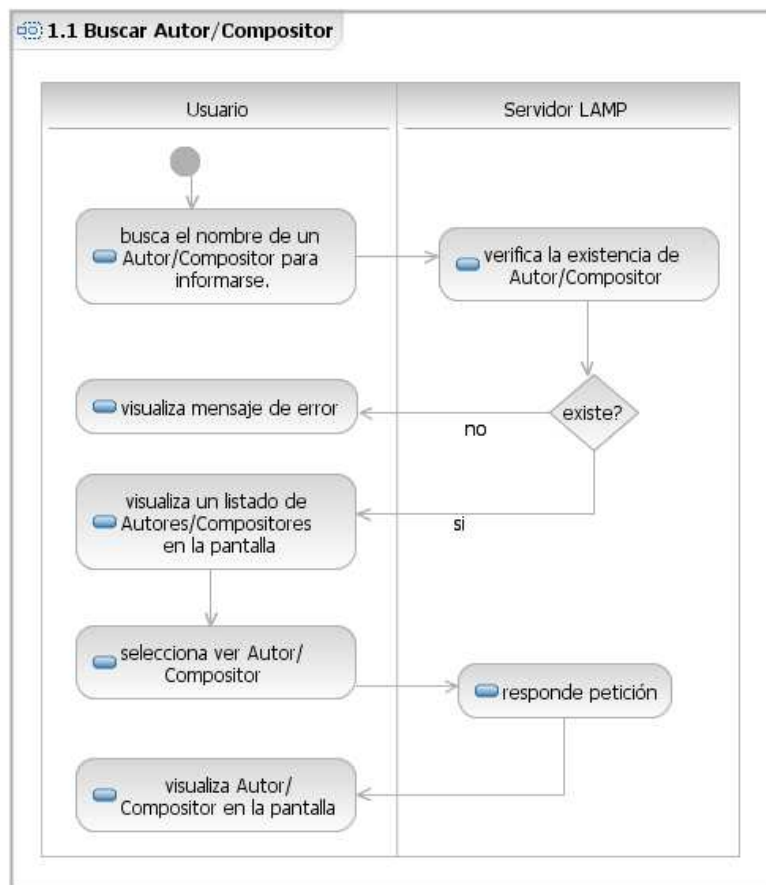
- 1 Informática busca el nombre del Operador para administrar.
- 2 El servidor responde petición.
- 3 Informática visualiza listado de operadores.
- 4 Informática verifica la existencia de operador.
- 5 *No existe* Operador:
 - 5.1 Informática selecciona agregar Operador.
- 5 *Si existe* Operador:
 - 5.2 Informática selecciona editar Operador.
 - 5.3 Informática selecciona eliminar Operador.

Parte VI: Diagramas de Actividad del Sistema – Sea Level

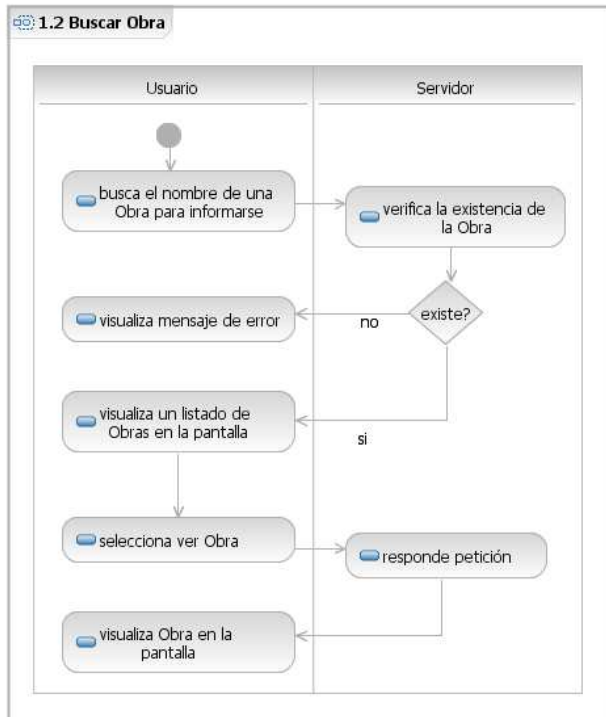
1 Buscar en el Catálogo de Autor/Compositor, Obra, Biografía



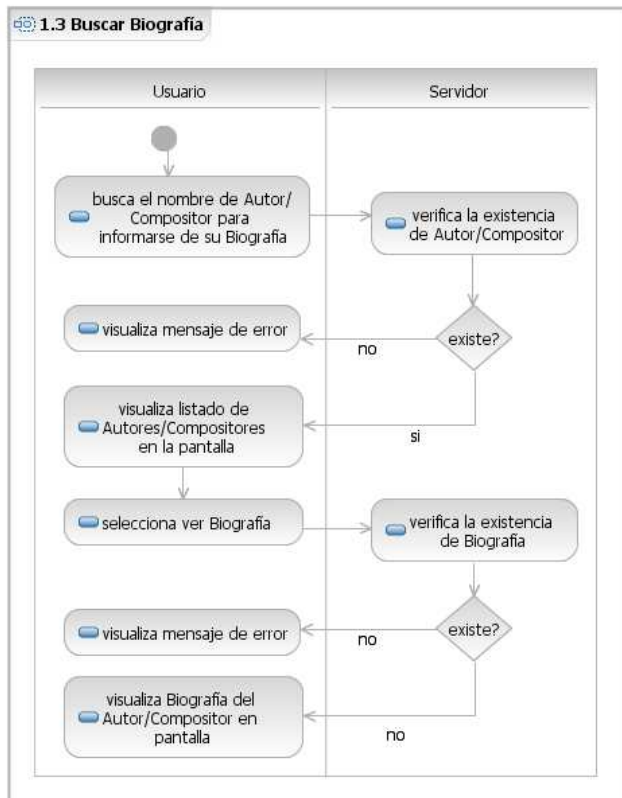
1.1 Buscar Autor/compositor



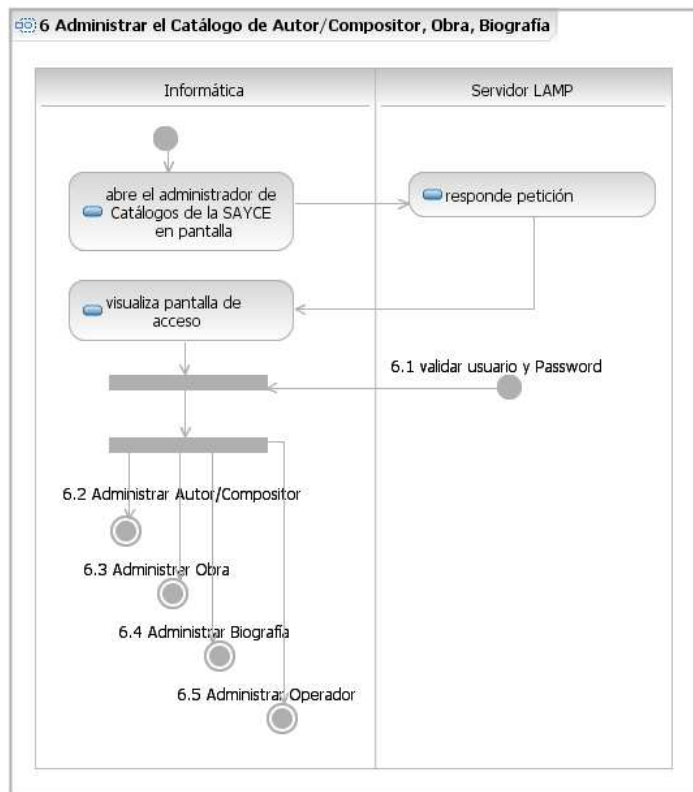
1.2 Buscar Obra



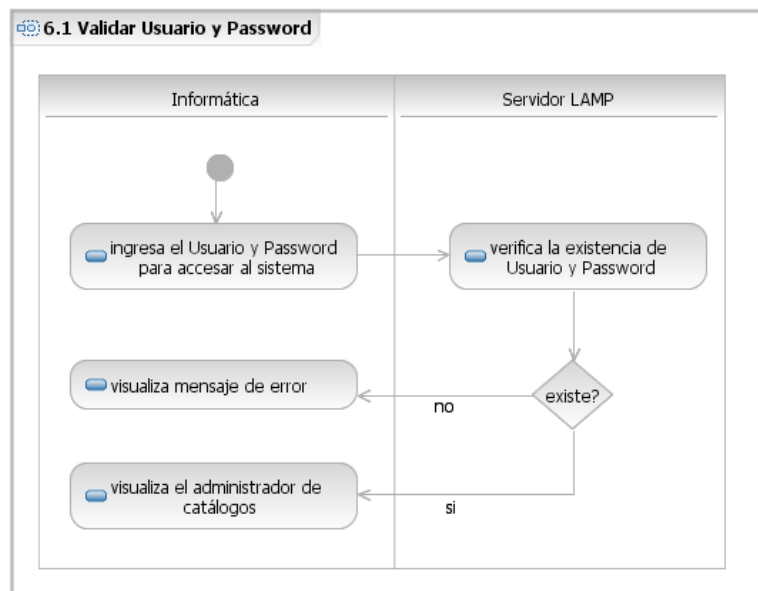
1.3 Buscar Biografía



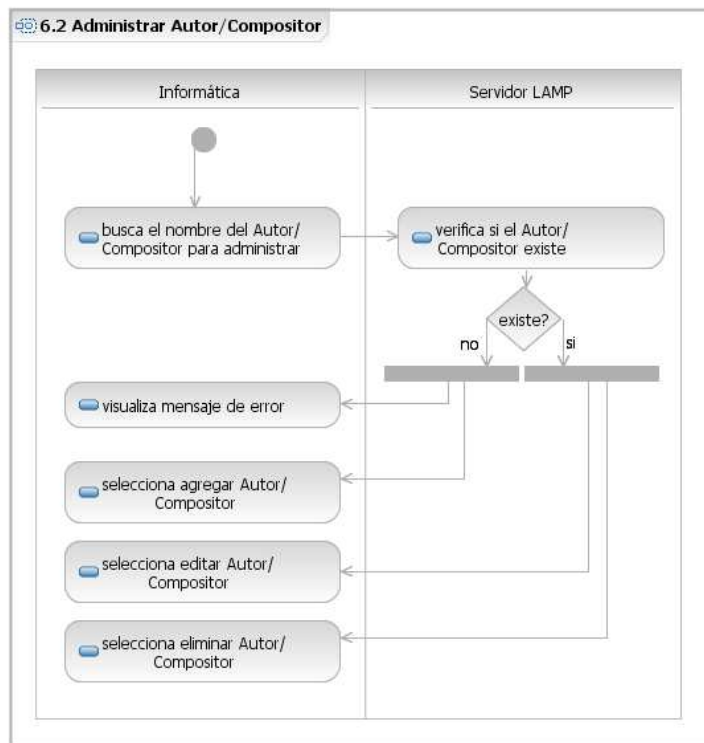
6 Administrar el Catálogo de Autor/Compositor, Obra, Biografía



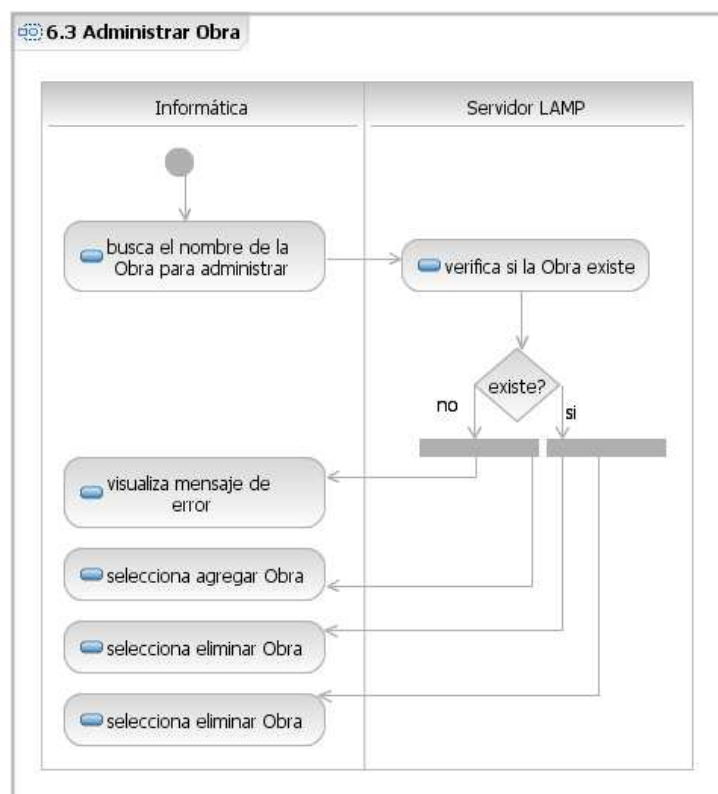
6.1 Validar Usuario y Password



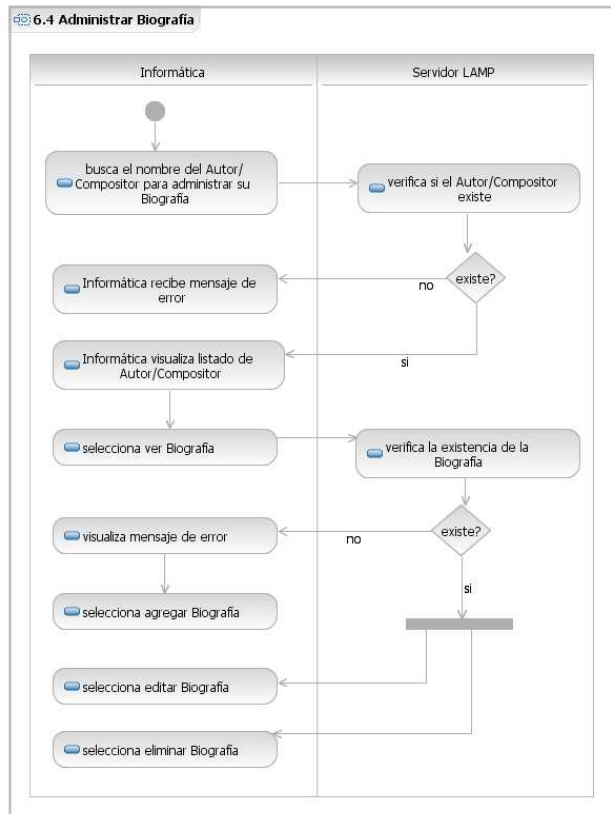
6.2 Administrar Autor/Compositor



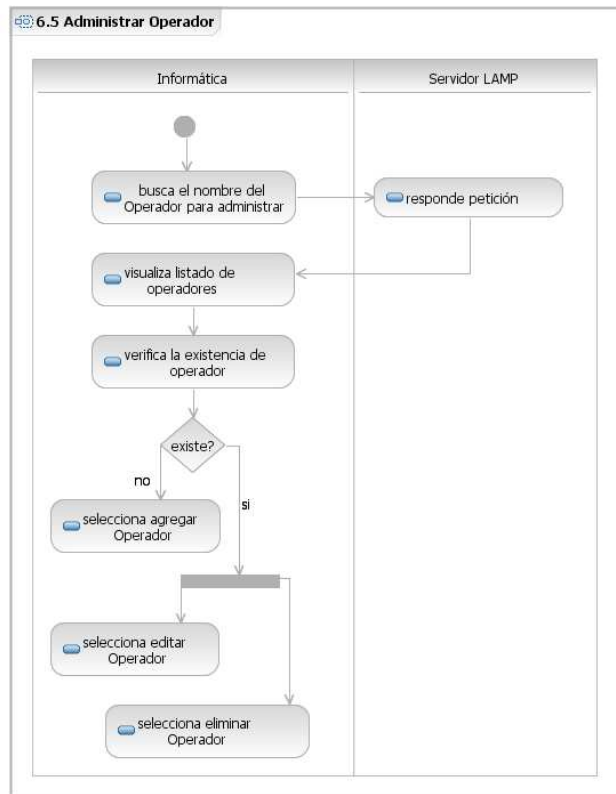
6.3 Administrar Obra



6.4 Administrar Biografía



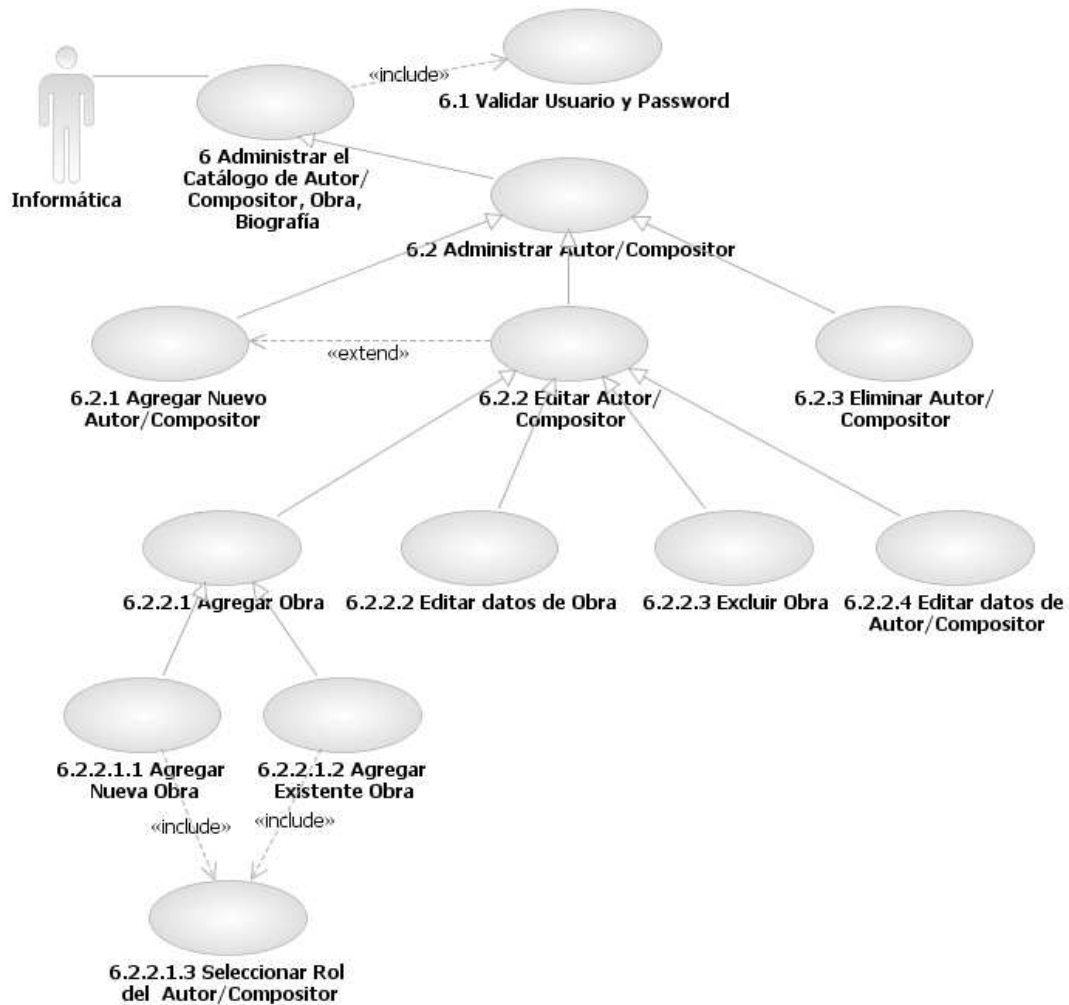
6.5 Administrar Operadores

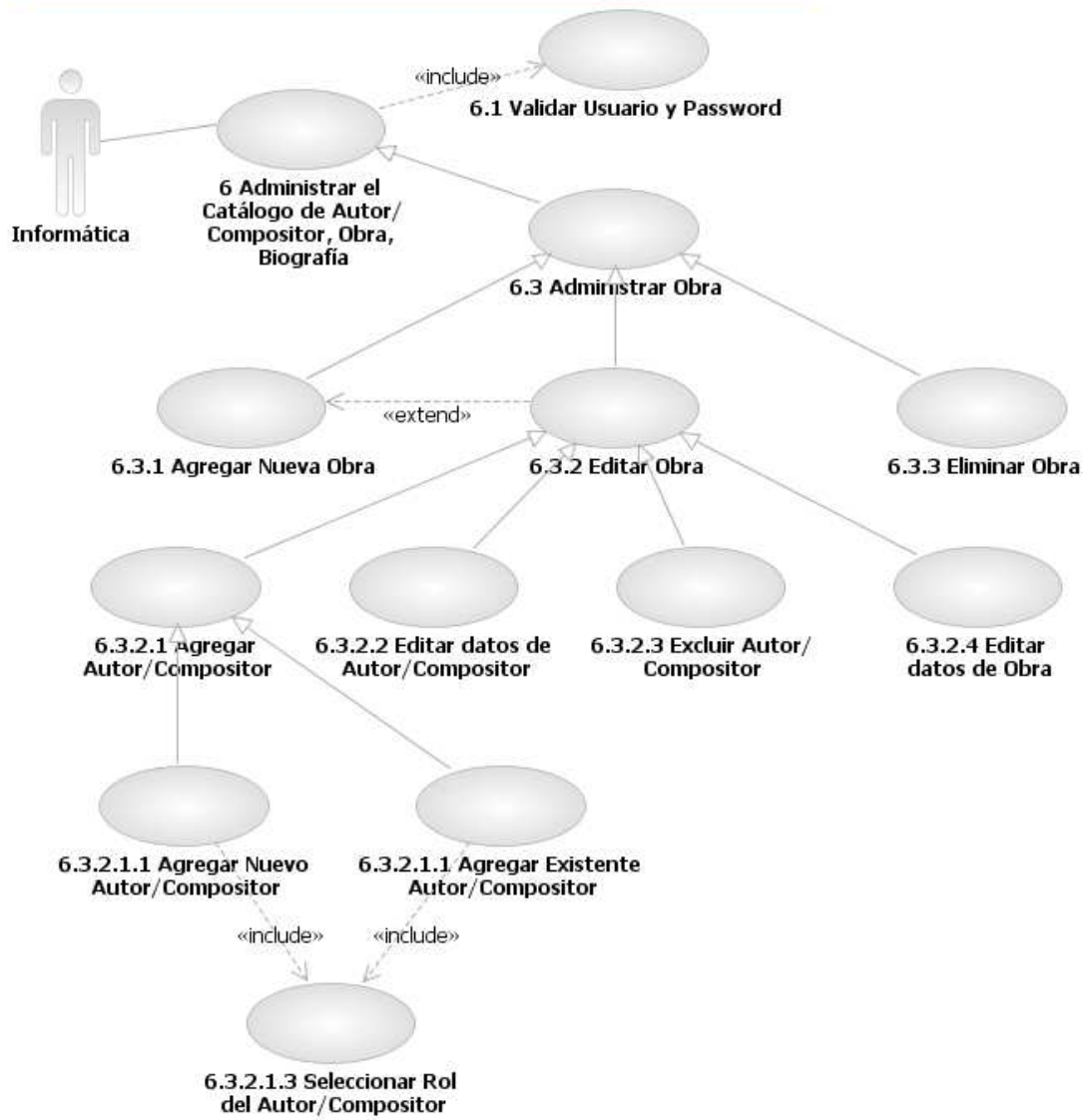


MODELO DEL SISTEMA – FISH LEVEL

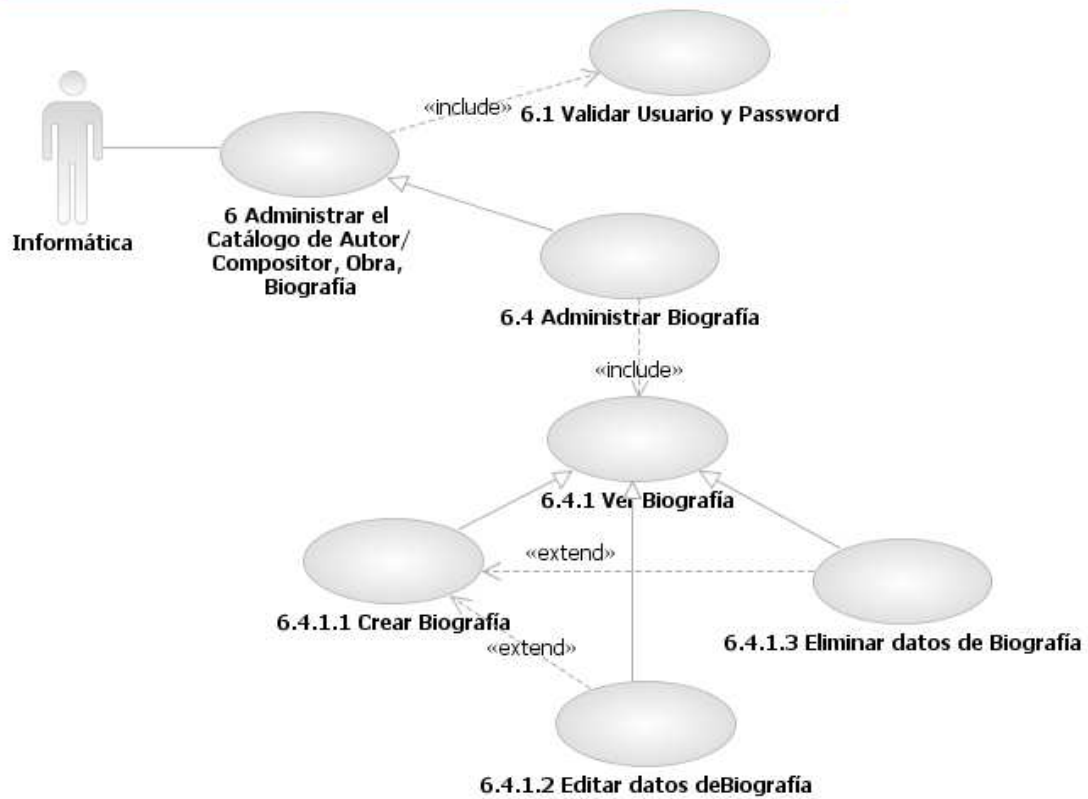
Parte VII: Diagrama de casos de uso del Sistema - Fish Level

Parte VII: Diagrama de Casos De Uso Del Sistema (Fish Level) : 6.2 Administrar Autor/Compositor

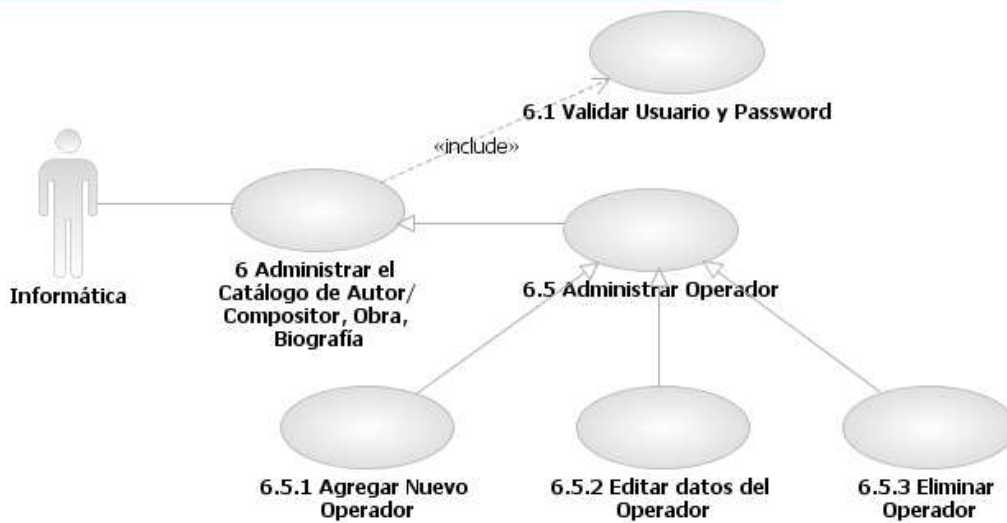




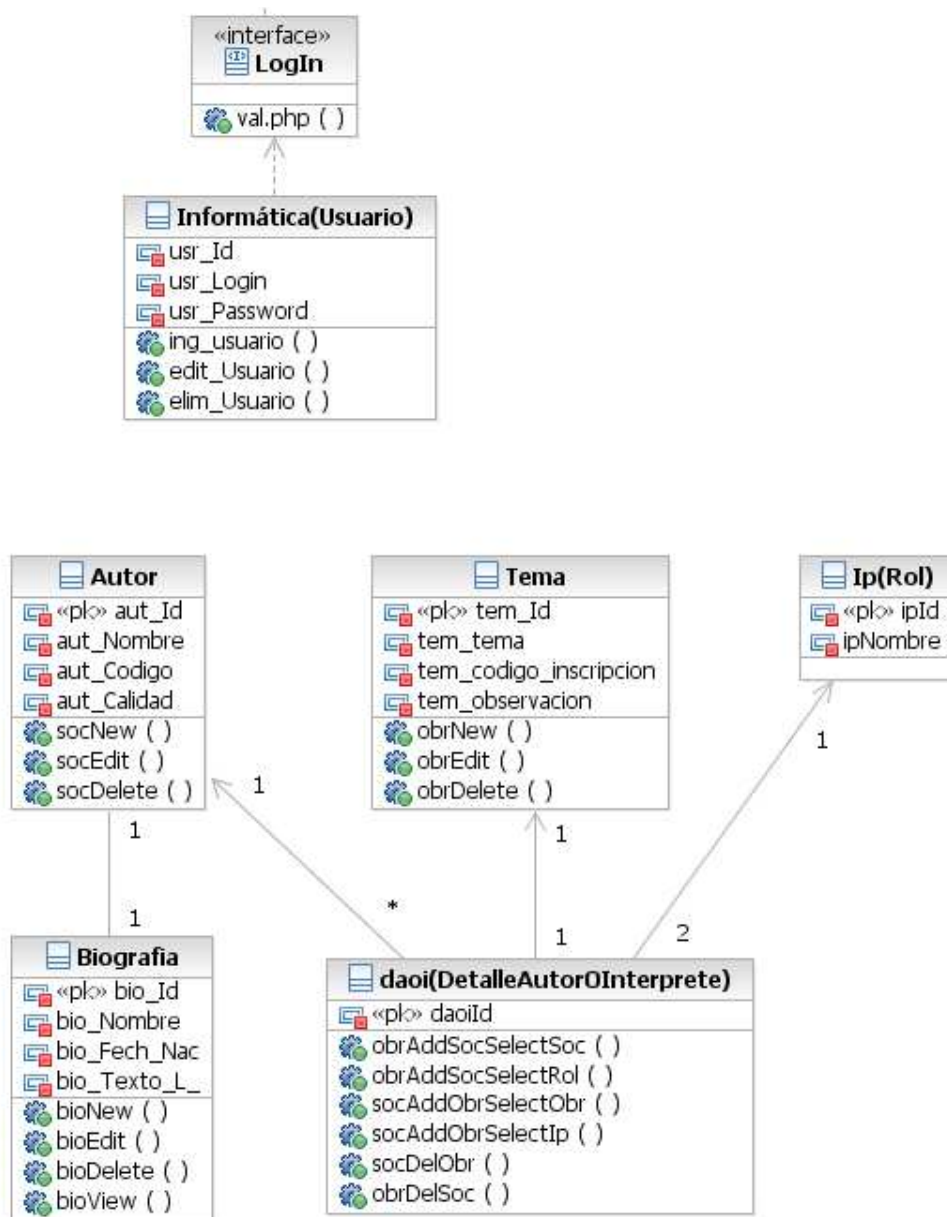
Parte VII: Diagrama de Casos De Uso Del Sistema (Fish Level) : 6.4 Administrar Biografía



Parte VII: Diagrama de Casos De Uso Del Sistema (Fish Level) : 6.5 Administrar Operador

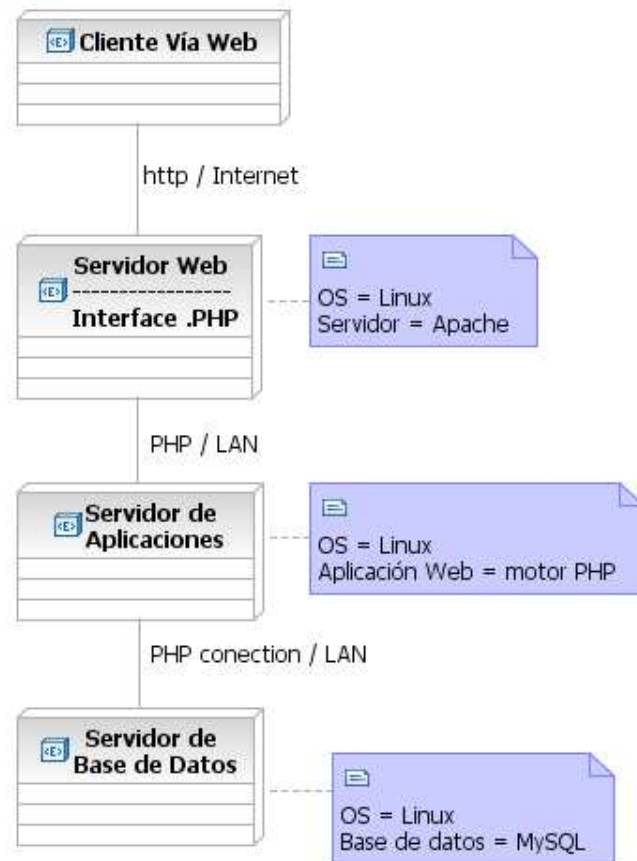


Parte VIII: Diagrama de clases del Sistema – Fish Level



Parte IX: Diagrama de implantación – Fish Level

Parte IX: Diagrama de implantación - Fish Level



Parte X: Diagrama de implantación – Fish Level

