

# **Aplicación del paradigma de aprendizaje práctico al proyecto E-LANE en la Universidad del Cauca mediante el desarrollo de laboratorios virtuales**



**ÁLVARO EFRÉN CORTÉS CASTRO  
SEGUNDO EDIVER CAICEDO DAZA**

Monografía presentada como requisito para optar el título de Ingeniero en  
Electrónica y Telecomunicaciones

Director: Ing.Esp. Mario Fernando Solarte

**Universidad del Cauca**  
**Facultad de Ingeniería Electrónica y Telecomunicaciones**  
**Departamento de Telemática**  
Popayán, Abril de 2006

## TABLA DE CONTENIDO

<u>INTRODUCCIÓN</u>	<u>¡ERROR! MARCADOR NO DEFINIDO.</u>
<u>I. BASE CONCEPTUAL</u>	<u>¡ERROR! MARCADOR NO DEFINIDO.</u>
1.1 E-LEARNING	¡ERROR! MARCADOR NO DEFINIDO.
1.2 PARADIGMAS DE FORMACIÓN (LEARNING BY READING AND LEARNING BY DOING)	¡ERROR! MARCADOR NO DEFINIDO.
1.2.1 SISTEMAS DE APRENDIZAJE PRÁCTICO BASADOS EN ACCESO REMOTO A SISTEMAS REALES	¡ERROR! MARCADOR NO DEFINIDO.
1.2.2 SISTEMAS DE APRENDIZAJE PRÁCTICOS BASADOS EN LABORATORIOS VIRTUALES	¡ERROR! MARCADOR NO DEFINIDO.
1.3 PROYECTO E-LANE ( EUROPEAN AND LATIN AMERICAN NEW EDUCATION)	¡ERROR! MARCADOR NO DEFINIDO.
1.4 SISTEMAS DE GESTION DE APRENDIZAJE (LMS)	¡ERROR! MARCADOR NO DEFINIDO.
1.5 PLATAFORMA DOTLRN	¡ERROR! MARCADOR NO DEFINIDO.
1.5.1 DOTLRN USER PORTAL	¡ERROR! MARCADOR NO DEFINIDO.
1.5.2. DOTLRN COMMUNITIES	¡ERROR! MARCADOR NO DEFINIDO.
1.5.3 PORTLETS	¡ERROR! MARCADOR NO DEFINIDO.
1.5.4 DOTLRN USERS	¡ERROR! MARCADOR NO DEFINIDO.
1.5.5 MÓDULOS DE DOTLRN:	¡ERROR! MARCADOR NO DEFINIDO.
<u>II. ARQUITECTURA PROPUESTA</u>	<u>¡ERROR! MARCADOR NO DEFINIDO.</u>
2.1 MODELADO DE REQUISITOS ESCENCIALES	¡ERROR! MARCADOR NO DEFINIDO.
2.2 MODELO DE CASOS DE USO	¡ERROR! MARCADOR NO DEFINIDO.
2.3 MODELO DE ANÁLISIS	¡ERROR! MARCADOR NO DEFINIDO.
2.4 ARQUITECTURA DE REFERENCIA	¡ERROR! MARCADOR NO DEFINIDO.
2.4.1 OPENACS-CORE	¡ERROR! MARCADOR NO DEFINIDO.
2.4.2 CAPA DE SERVICIOS	¡ERROR! MARCADOR NO DEFINIDO.
2.4.3 CAPA DE APLICACIONES	¡ERROR! MARCADOR NO DEFINIDO.
2.4.4 SERVICIOS DE LABORATORIOS VIRTUALES	¡ERROR! MARCADOR NO DEFINIDO.
<u>III. DISEÑO DE UN PROTOTIPO DE LABORATORIO VIRTUAL.</u>	<u>¡ERROR! MARCADOR NO DEFINIDO.</u>
3.1 DESCRIPCIÓN FUNCIONAL DEL LABORATORIO VIRTUAL AXE-10	¡ERROR! MARCADOR

**NO DEFINIDO.**

3.2 MODELO DEL AMBIENTE	¡ERROR! MARCADOR NO DEFINIDO.
3.2.1 DESCRIPCIÓN DE ACTORES	¡ERROR! MARCADOR NO DEFINIDO.
3.3 MODELO DE ANÁLISIS DEL SISTEMA	¡ERROR! MARCADOR NO DEFINIDO.
3.3.1 DIAGRAMAS DE CLASES DEL SERVICIO	¡ERROR! MARCADOR NO DEFINIDO.
3.3.2 DESCRIPCIÓN DE PAQUETES Y CLASES DE ANÁLISIS DEL SISTEMA.	¡ERROR! MARCADOR NO DEFINIDO.
3.3.3. DIAGRAMAS DE COLABORACIÓN Y/O SECUENCIA PARA LOS CASOS DE USO DEL SISTEMA.	¡ERROR! MARCADOR NO DEFINIDO.
3.4 MODELO DE DISEÑO DEL SISTEMA.	¡ERROR! MARCADOR NO DEFINIDO.
3.4.1 ARQUITECTURA DE REFERENCIA PARA EL SISTEMA.	¡ERROR! MARCADOR NO DEFINIDO.
3.4.2 DESCRIPCIÓN DE SUBSISTEMAS.	¡ERROR! MARCADOR NO DEFINIDO.
3.4.3 DESCRIPCIÓN DE LAS CLASES DE DISEÑO.	¡ERROR! MARCADOR NO DEFINIDO.
3.5. MODELO DE DESPLIEGUE DEL SISTEMA	¡ERROR! MARCADOR NO DEFINIDO.
3.6 INTERFACES GRAFICAS DEL PROTOTIPO FINAL	¡ERROR! MARCADOR NO DEFINIDO.

**IV. CONCLUSIONES Y RECOMENDACIONES** ¡ERROR! MARCADOR NO DEFINIDO.

**V. REFERENCIAS BIBLIOGRAFICAS** ¡ERROR! MARCADOR NO DEFINIDO.

## LISTA DE FIGURAS.

Figura 1.1 Arquitectura del e-lab .....	9
Figura 1.2 Laboratorio virtual de procesamiento de señal.....	12
Figura 1.3 Ejemplo de una sesión cliente/servidor en el sistema CROS.....	13
Figura 1.4 SimulNet on the World Wide Web.....	16
Figura 1.5 Arquitectura Distribuida de Simulnet.....	17
Figura 1.6 Arquitecturas OpenACS y dotLRN.....	28
Figura 1.7 dotLRN User Portal .....	29
Figura 1.8 Diagrama de Clases para dotLRN .....	30
Figura 1.9 Paquete de clases portlets dotLRN.....	31
Figura 1.10 Paquete parties OpenACS.....	32
Figura 1.11 Paquete de Usuario dotLRN.....	32
Figura 2.1. Diagrama de Casos de Uso Genérico.....	44
Figura 2.2. Diagrama Extendido del Caso de Uso Gestionar Sistema.....	46
Figura 2.3 Diagrama Extendido del Caso de Uso “Asesorar Estudiante”.....	47
Figura 2.4. Diagrama Extendido del Caso de Uso “Realizar Práctica”.....	48
Figura 2.5. Diagrama acs-subsite.....	49
Figura 2.6 Diagrama portlet.....	51
Figura 2.7 Diagrama dotLRN-user.....	52
Figura 2.8. Arquitectura de referencia para la implementación de laboratorios virtuales en dotLRN .....	53
Figura 2.9. Core OpenACS .....	55
Figura 3.1 Servicio de Práctica - Gestión de Servicios Especiales, Visualización de Componentes y Seguimiento de Prácticas.....	68
Figura 3.2 Diagrama de clases para el Servicio Ejecutar Práctica – Visualización de Componentes .....	71
Figura 3.3 Diagrama de clases para el Servicio Ejecutar Práctica – Gestión de Servicios Especiales .....	72
Figura 3.4 Modelo de Datos para el Laboratorio Virtual AXE .....	74
Figura 3.5 Maquina de estados finitos para la entrega de la Práctica de Gestión de Servicios Especiales del Laboratorio Virtual AXE .....	75
Figura 3.6 Diagrama de clases para el Servicio Ejecutar Práctica – Visualización de Seguimiento .....	76
Figura 3.7 Paquetes de la Aplicación Laboratorio Virtual AXE .....	77
Figura 3.8 Diagrama de colaboración para el escenario Visualización de Componentes.....	80
Figura 3.9 Diagrama de colaboración para el escenario Gestión de Servicios Especiales.....	82
Figura 3.10 Diagrama de colaboración para el caso de uso Ver Seguimiento .....	83
Figura 3.11 Arquitectura Distribuida para el Laboratorio Virtual AXE-10.....	84
Figura 3.12 Arquitectura Básica .....	88
Figura 3.13 Interfaz de validación de ingreso.....	90
Figura 3.14 Interfaz Class Home .....	91
Figura 3.15 Interfaz de Presentación .....	92
Figura 3.16 Interfaz de Selección de Práctica.....	93
Figura 3.17 Interfaz de Conexión del Laboratorio Virtual.....	93
Figura 3.18 Interfaz Principal de Práctica de Servicios Especiales.....	94
Figura 3.19 Interfaz de Selección de Abonados Disponibles.....	95
Figura 3.20 Interfaz Principal de Práctica de Servicios Especiales.....	96
Figura 3.21 Interfaz de Ejecución .....	97
Figura 3.22 Interfaz Principal de Práctica de Servicios Especiales.....	98
Figura 3.23 Interfaz de Validación .....	98
Figura 3.24 Interfaz de Visualización de Componentes .....	99
Figura 3.25 Interfaz de Visualización de Componentes .....	100
Figura 3.26 Interfaz de Seguimiento de Prácticas.....	101
Figura 3.27 Interfaz de Actividades de Prácticas .....	102

## RESUMEN

Las posibilidades ofrecidas por la tecnología han permitido la aparición de múltiples sistemas de aprendizaje basados en computador. Internet ha sido utilizado con éxito como medio para la creación de entornos virtuales de aprendizaje. Diversas instituciones públicas y privadas hacen uso de las nuevas tecnologías para proveer de programas de aprendizaje y formación en los diferentes niveles, tanto para la enseñanza teórica como para el aprendizaje práctico. Existen multitud de sistemas de ambos tipos, con características diferentes, desarrollados sobre diferentes plataformas. Estos sistemas de aprendizaje están fundamentados en dos paradigmas, según sea su tipo: **Learning by reading** y **Learning by doing**.

El primer paradigma es implementado principalmente a través de plataformas de aprendizaje denominadas Sistemas de Gestión de Aprendizaje (LMS), las cuales han sido ampliamente adoptadas por universidades y organizaciones, con el objetivo de dar respuesta a las demandas que en materia de educación se le exige. Sin embargo estos Sistemas de Gestión de Aprendizaje (LMS) dejan de lado las ventajas y beneficios del paradigma learning by doing.

La Universidad del Cauca en el marco de los procesos de mejora de la calidad de la docencia, fomento a la investigación y descentralización, participa actualmente en el proyecto E-LANE (European and Latin-American New Education). En este contexto y con el objetivo de contar con una oferta académica de cursos disponibles a través de la plataforma de e-learning dotLRN, la Universidad del Cauca implementará algunos cursos que podrán ofrecerse en “modalidad virtual”. Muchos cursos de los diversos programas que la universidad ofrece requieren de la práctica como un elemento sustancial para el aprendizaje ya que la interactividad con el sistema permite una gran comprensión del contenido teórico aprendido, situación que justifica la necesidad de implementar sistemas de tele-educación prácticos con el objetivo de brindar apoyo al

desarrollo normal de estos cursos, experiencia que pueda replicarse en otras iniciativas y disciplinas de similar propósito. Sin embargo la plataforma dotLRN, no tiene contemplado el soporte al paradigma de aprendizaje práctico, aspecto de significativo valor agregado dentro del ámbito de la educación virtual.

La búsqueda de tal solución inició con la descripción de las aproximaciones del paradigma learning by doing, en la cual se fija una caracterización de los laboratorios virtuales más representativos que lo implementan; características que se describieron en términos de las funcionalidades, arquitecturas de referencia y la base tecnológica que ellos presentan en su conjunto. Esta bibliografía permitió establecer las funcionalidades que un laboratorio debe proveer enmarcadas dentro de las características fundamentales del e-learning, (comunicación, contenido, y seguimiento de las actividades del estudiante), como también las posibilidades que un laboratorio virtual puede llegar a proveer al estar integrado a un LMS en términos de gestión y distribución.

La exploración tecnológica de la plataforma dotLRN, permitió establecer las fortalezas relativas al desarrollo de aplicaciones web orientadas a comunidades, tanto en el nivel de aplicaciones como en el de servicios reutilizables en la aplicación del aprendizaje práctico.

La exploración teórica y tecnológica permitió establecer una arquitectura de referencia propuesta para la implementación de laboratorios virtuales basada en la misma arquitectura de dotLRN, con la cual se busca la construcción de laboratorios virtuales, fundamentada en la reutilización de los diversos servicios e-learning que ofrece y que permita la integración de diversos subsistemas de aprendizaje, así como la entrega adecuada de los mismos a los estudiantes.

Finalmente con la implementación del prototipo Laboratorio Virtual AXE-10 integrado al LMS dotLRN, no sólo se logró la validación de la arquitectura propuesta, sino conocer de manera general las ventajas y desventajas de OpenACS y su LMS.

## INTRODUCCIÓN

El rápido desarrollo de las tecnologías de la información no solo ha cambiado dramáticamente adquisición, manipulación y transmisión de conocimiento, sino que ha llevado a un cambio en los paradigmas de educación como consecuencia de las nuevas demandas sociales que se hacen de manera continua. Lo anterior ha llevado al surgimiento de nuevos modelos educativos centrados en el estudiante, diferentes del sistema tradicional, el cual se fundamenta en clases magistrales impartidas por los profesores. En tales modelos el estudiante tiene una reducción significativa de horas presenciales y una ampliación de las actividades participativas, en dicho sentido se busca formar profesionales con poco tiempo disponible y con un elevado rendimiento de su tiempo.

En este orden de ideas el proyecto “Aplicación del paradigma learning by doing mediante la construcción de laboratorios virtuales en el proyecto E-LANE, en la Universidad del Cauca”, constituye un acercamiento hacia nuevos modelos, enmarcados en los nuevos cambios que exige la sociedad en términos de demandas educativas, sociales y tecnológicos.

La ejecución de este proyecto, aporta al desarrollo académico de la FIET al explorar un nuevo campo del aprendizaje disminuyendo la brecha existente entre los sistemas de aprendizaje convencionales y los sistemas de aprendizaje práctico dentro del ámbito de la educación virtual; de esta manera el presente trabajo se constituye en un esfuerzo tendiente hacia una mayor popularización de tales sistemas en la Universidad del Cauca, en la región, en el país, y en la comunidad académica en general.

El trabajo está estructurado en cinco capítulos, de los cuales el capítulo uno corresponde al estado del arte del aprendizaje práctico, cuyo análisis realizado establece una separación entre los sistemas que entregan recursos educativos orientados al aprendizaje teórico y aquellos orientados al aprendizaje práctico.

El capítulo dos se concentra en los laboratorios virtuales como aplicaciones e-learning, con el propósito de conocer sus principales elementos y sus características, para poder establecer los criterios que describen el contenido de un laboratorio, así como los elementos para su distribución. Este capítulo contiene las abstracciones genéricas, características y aspectos fundamentales a tener en cuenta en la implementación del paradigma learning by doing en cualquier tipo de plataforma e-learning, que son aplicables a la plataforma dotLRN.

El capítulo tres, presenta el diseño e implementación del prototipo de un laboratorio virtual integrado al LMS dotLRN, con el objetivo de validar la arquitectura propuesta del paradigma de aprendizaje práctico que se propone en el capítulo dos. Este capítulo contiene las diferentes decisiones de diseño, implementación y desarrollo tomadas en la realización del proyecto, tales como selección de la arquitectura de la aplicación, modelos descriptivos, diagramas correspondientes, y tecnologías empleadas.

El capítulo cuatro contiene las conclusiones respectivas, producto de la realización del proyecto, tanto a nivel del desarrollo como a nivel teórico y de utilización de herramientas, de igual manera, contiene las recomendaciones producto de la experiencia adquirida, que puedan ser aplicadas en el desarrollo de trabajos posteriores.

El capítulo cinco contiene la referencia a los documentos que fueron consultados como fuente de conocimiento en el desarrollo del proyecto, así como una lista de los términos que deben ser tenidos en cuenta para un mejor entendimiento de la documentación, producto también generado en el desarrollo del proyecto.

Finalmente los anexos contienen la descripción de cada uno de los documentos generados con el objetivo de profundizar o esclarecer un tema específico.



La metodología de referencia utilizada fue el Modelo para Construcción de Soluciones (MCS) presentado en el Modelo Integral para un Profesional en Ingeniería [SC02]. Este modelo describe el proceso de generación de nuevo conocimiento, tiene como paradigma principal la técnica de desarrollo incremental y se concibe como referencia metodológica esencial para cualquier proyecto cuyo propósito sea construir una solución de calidad, oportuna y con costos competitivos.

## **I. BASE CONCEPTUAL**

En el marco de las nuevas demandas educativas de la sociedad de América latina, la aplicación del paradigma learning by doing al proyecto E-LANE, se constituye en un valioso aporte, por cuanto permite generar contenidos de gran valor agregado en materia académica, buscando así, mejorar la calidad y los resultados de los procesos de aprendizaje.

En este orden de ideas el presente capítulo busca ser el punto de partida en materia académica mediante la descripción del estado del arte del paradigma learning by doing, así como ser un soporte teórico - práctico a la aplicación del mismo al proyecto E-LANE mediante la implementación de laboratorios virtuales en la plataforma dotLRN de OpenACS, propuesta que se expone en el capítulo siguiente.

El capítulo contiene la descripción del paradigma de formación learning by doing, con sus dos aproximaciones más sobresalientes; los sistemas de aprendizaje práctico basados en acceso remoto a sistemas reales y los sistemas de aprendizaje prácticos basados en laboratorios virtuales; exposición que se hace en términos de sus funcionalidades más representativas y cuya consideración fundamentarán la aplicación del paradigma learning by doing al proyecto E-LANE. La exposición del proyecto E-LANE y sus objetivos, junto con las características y componentes de un LMS son el marco de referencia para la exploración y descripción de las funcionalidades generales del LMS dotLRN.

### **1.1 E-LEARNING**

Podemos definir el e-learning como el suministro de programas educacionales y sistemas de aprendizaje a través de medios electrónicos. El e-learning se basa en el uso de una computadora u otro dispositivo electrónico para proveer a las personas de material educativo [GP04].

La educación a distancia creó las bases para el desarrollo del e-learning, el cual viene a resolver algunas dificultades en cuanto a tiempos, sincronización de agendas, asistencia y viajes, problemas típicos de la educación tradicional.

El término e-learning abarca un amplio paquete de aplicaciones y procesos, como el aprendizaje basado en web, salones de clases virtuales y colaboración. No existe una definición precisa de este término. Clark Aldrich define un LMS como una amplia combinación de procesos, contenidos, e infraestructura para usar computadores y redes para escalar o mejorar una o más partes significativas de una cadena de aprendizaje, incluyendo gestión y distribución [AC02].

Técnicamente, el e-learning es la entrega de material educativo vía cualquier medio electrónico, incluyendo Internet, intranets, CD, DVD, entre otros medios. Para los educadores, el e-learning es el uso de tecnologías de redes y comunidades para diseñar, seleccionar, administrar, entregar y extender la educación.

Siendo breves y prácticos, el e-learning es el aprendizaje basado en tecnología. Siendo descriptivos la educación electrónica es la capacitación y adiestramiento de estudiantes y empleados usando materiales disponibles para web a través del Internet, llegando a ofrecer sofisticadas facilidades como flujo de audio y video, presentaciones, vínculos a información relativa al tema publicado en la web, animación, libros electrónicos, y aplicaciones para la generación y edición de imágenes [PG04].

En este orden de ideas, los avances que hemos presenciado en las tecnologías de la Información y comunicaciones, y más específicamente en multimedia, tecnologías de red e ingeniería del software permiten prever la aparición de una nueva generación de sistemas de aprendizaje basados en computador. Internet es, hoy por hoy, el entorno de soporte más apropiado para los sistemas de aprendizaje distribuidos. Como consecuencia de ello, tanto las instituciones académicas públicas como las privadas, aprovechan los avances de la tecnología para ofrecer a sus estudiantes, productos y

servicios de aprendizaje de calidad superior sobre el World Wide Web. En esta situación, los sistemas educativos electrónicos son cada vez más numerosos y se pueden clasificar en sistemas de aprendizaje teórico y sistemas de aprendizaje práctico [AF01].

## **1.2 PARADIGMAS DE FORMACIÓN (LEARNING BY READING AND LEARNING BY DOING)**

Los sistemas de aprendizaje teórico, tienen como objetivo proporcionar plataformas para la gestión y entrega de materiales educativos de forma que los estudiantes puedan acceder a ellos a través de un navegador Web. Estos recursos educativos son principalmente expositivos, siguiendo el paradigma learning by reading [AF01], y su objetivo global es ofrecer entornos de aula virtual.

Los sistemas de aprendizaje práctico van más allá, ofrecen mecanismos para poner en práctica los conocimientos teóricos adquiridos, por ejemplo, con simulaciones, acceso remoto a laboratorios, etc. El objetivo final de tales sistemas es proporcionar entornos de laboratorio virtual, equivalentes a los laboratorios docentes y talleres prácticos de los sistemas educativos y de formación convencionales. El número de sistemas que implementan el concepto de laboratorio virtual sobre Internet es mucho menor que aquellos que ofrecen entornos de aulas virtuales.

Es posible distinguir dos aproximaciones diferentes en aquellos sistemas que ofrecen entornos de aprendizaje práctico sobre Internet: sistemas que ofrecen acceso remoto al equipamiento del laboratorio real y soluciones basadas en el uso de simuladores. En el primer tipo, los estudiantes utilizan una interfaz de acceso remoto a sistemas de control del equipo existente en los laboratorios. Los estudiantes utilizan, en el entorno de aprendizaje electrónico, los mismos recursos educativos que son usados en el laboratorio convencional, tanto software como hardware. La segunda aproximación al aprendizaje electrónico práctico utiliza simuladores de herramientas, equipos o procesos que son objeto de estudio en el laboratorio real.

### **1.2.1 Sistemas de aprendizaje práctico basados en acceso remoto a sistemas reales**

La idea principal en la que se basan los sistemas que proporcionan acceso remoto a equipamiento real a través de Internet es la de ofrecer, habitualmente vía un navegador Web, medio a través del cual los estudiantes pueden enviar comandos a un servidor, ejecutarlos en el laboratorio real, recibir los resultados de dichos comandos y presentarlos convenientemente.

Este tipo de sistemas debe proporcionar herramientas que permitan a los estudiantes seguir la evolución en el laboratorio de los experimentos. Además es necesario proveer mecanismos de seguridad que eviten situaciones de peligro para que los equipos utilizados debido a su uso por parte de aprendices inexpertos.

A su vez, dentro de los sistemas que ofrecen mecanismos de acceso remoto al equipo del laboratorio real, se puede realizar una distinción entre aquellos laboratorios cuyo equipo es hardware y aquellos laboratorios puramente software.

#### **a) Laboratorios basados en hardware**

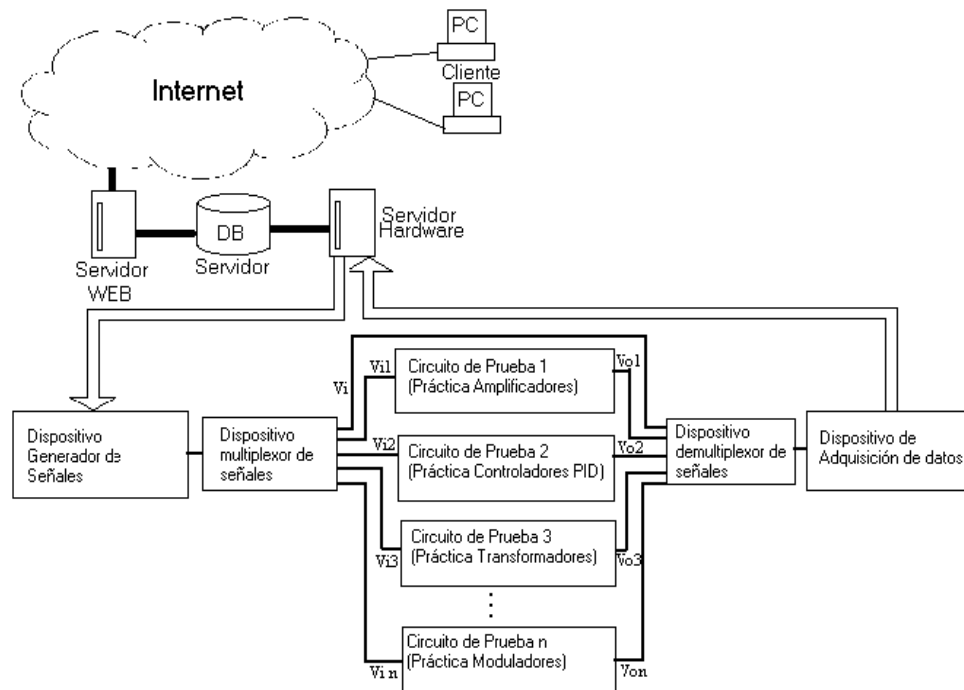
Existen muchas áreas en las que es particularmente interesante la utilización de aquellos instrumentos, herramientas o equipos que los estudiantes encontrarán en el mercado. En algunas áreas, el aprendizaje teórico es claramente insuficiente si no viene acompañado por la puesta en práctica de los conocimientos adquiridos mediante el manejo y práctica con los equipos objeto de estudio. Algunos de estos campos son sistemas mecánicos, sistemas de alta potencia, etc. Para este tipo de laboratorios, existen plataformas que permiten controlar de forma remota el equipo existente en el laboratorio, con las mismas posibilidades que se tendrían si los estudiantes estuviesen físicamente en el laboratorio.

Entre los sistemas que siguen esta filosofía se puede citar a Second Best to Being There [AF01], desarrollado en la universidad de Oregon State de los EEUU. SBBT proporciona acceso remoto a un laboratorio de ingeniería de control en el que los estudiantes pueden manejar instrumentos reales (robots, sistemas de suspensión magnética, motores, etc.) desde sus casas. Los estudiantes tienen control absoluto sobre el entorno y pueden llevar a cabo todos los experimentos que podrían si estuviesen en el laboratorio.

SBBT está construido utilizando el estilo arquitectónico cliente servidor sobre el protocolo UDP/IP. La telepresencia de los estudiantes en el laboratorio se consigue gracias a una interfaz de usuario con audio, video y sistema de adquisición de datos. Una pizarra virtual sirve como soporte para la colaboración entre los estudiantes. La seguridad de los experimentos está garantizada mediante un sistema específico que evita situaciones de peligro en el laboratorio real.

En Colombia, la Universidad Nacional Sede Manizales desarrolló un primer prototipo de Laboratorio Virtual, denominado e-lab, Laboratorio Interactivo Remoto Sobre La WEB [CG01], orientado a las aplicaciones que empleen Internet como plataforma, y especialmente, las que se realicen para entornos educativos.

El laboratorio está diseñado sobre el modelo Cliente–Servidor, que determina la estructura general de todo el sistema, ver figura 1.1. El sistema cumple con los requerimientos de software tales como: mayor portabilidad y conectividad, régimen de trabajo multiusuario y multitarea, entre otros



**Figura 1.1 Arquitectura del e-lab**

El laboratorio virtual de medición vía Internet (e-Lab) cumple con las características convencionales de medición presencial, a la vez que permite mediciones a través de PC con todas sus ventajas, entre las cuales se tienen: proceso de datos en tiempo real y para análisis posterior, visualización de señales, configurabilidad por software, modificación de parámetros, lenguajes estándar de comunicaciones con los instrumentos de medición, infraestructura abierta, posibilidad de expansión, etc.

**b) Laboratorios basados en software**

Es común encontrar situaciones en las que los laboratorios tradicionales están basados en herramientas software: informática, bases de datos o ingeniería del software son algunos ejemplos. En este caso, la reutilización de experimentos sobre los paquetes software utilizados en el laboratorio convencional, se realiza una interfaz de acceso al paquete, a la cual se accede remotamente desde el equipo del estudiante.

Muchos de estos experimentos necesitan paquetes software de altas prestaciones disponibles únicamente en universidades o centros de investigación y que no están al alcance de los estudiantes. Por ello, en un entorno de aprendizaje electrónico a distancia, la solución que se ofrece a los estudiantes es el acceso remoto a los recursos existentes en el laboratorio mediante una interfaz adecuada. Habitualmente, la herramienta básica para el soporte de esta interfaz es un navegador Web, a través del cual pueden realizar todas las operaciones que estarían a su disposición si se encontrasen en el propio laboratorio.

Una de las propuestas más interesantes de este tipo de laboratorios es Real Experiment eXecution (REX), desarrollado en la universidad de Génova en Italia, [AF01]. En esta plataforma los experimentos se llevan a cabo mediante la ejecución de secuencias de operaciones. REX implementa interfaces software capaces, tanto de actuar sobre el experimento software mediante el envío de comandos, como de recoger resultados y presentarlos al estudiante. Estos resultados se acompañan de explicaciones y comentarios de los tutores. Es importante destacar el hecho de que los comandos se ejecutan, efectivamente, sobre un sistema real, lo que permite al estudiante probar todas las posibles opciones asociadas con el comando, incluyendo aquellas susceptibles de conducir el sistema a una situación de error o excepción, tal y como se produciría al utilizar el sistema en un entorno no docente.

### **1.2.2 Sistemas de aprendizaje prácticos basados en laboratorios virtuales**

Las simulaciones basadas en computador implican la representación dinámica de procesos, incorporando diferentes grados de interacción por parte del usuario. Para entornos educativos, una ventaja de la simulación es que permite resaltar los aspectos más importantes de los experimentos dejando a un lado aquellos que no son relevantes desde el punto de vista pedagógico. Además, pueden proporcionar a los estudiantes



mejores respuestas en función de cómo están utilizando el simulador en sí. Es decir, los simuladores pueden adaptar su comportamiento al de los estudiantes para guiar a éstos por el camino correcto. La simulación es controlada por el estudiante. Éste puede proporcionar valores para las variables de entrada y de esta forma modificar la salida. Mediante la simulación, los estudiantes pueden modificar el comportamiento de un proceso de la misma forma que harían sobre un sistema real.

La simulación es una solución valiosa en sistemas de aprendizaje basados en computador, entre otros, debido a los siguientes factores:

1. La simulación es una solución efectiva en costo, ya que necesita menos espacio físico y es más barata que el equipamiento real utilizado en entornos industriales.
2. Factores de seguridad. Los estudiantes inexpertos podrían dañar los sistemas utilizados en los experimentos.
3. Libertad en la experimentación. La incorporación de nuevos elementos o llevar los experimentos a situaciones extremas sería impensable sobre equipamiento real.

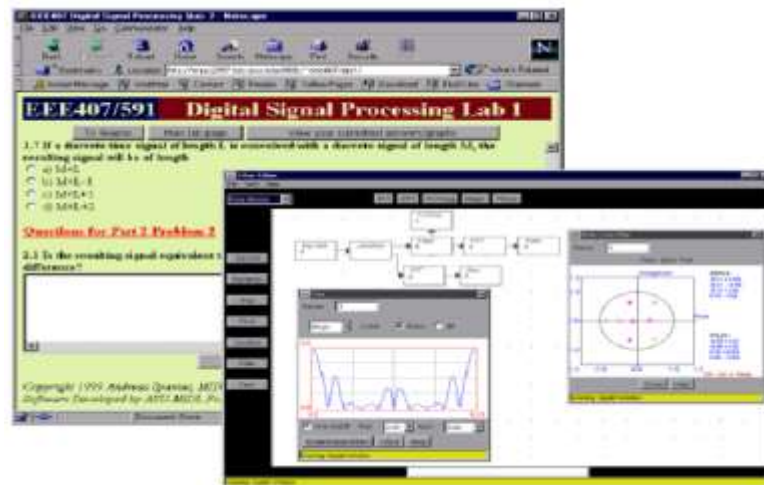
Los laboratorios virtuales sobre Internet que utilizan la simulación como elemento central para aprendizaje práctico, pueden clasificarse en dos grupos en función de su arquitectura [AF01]:

1. Ejecución en el cliente. Los simuladores de los instrumentos de un laboratorio real pueden ser ejecutados en los computadores de los estudiantes. La tecnología Java permite entregar, a través de Internet, software en forma de applets que se ejecutan en navegadores Web y que permiten a los estudiantes trabajar de forma interactiva con el simulador.
2. Ejecución en el servidor. Los simuladores se ejecutan en un servidor y son accesibles

de forma remota para los estudiantes a través de Web. El navegador Web se utiliza como interfaz entre los estudiantes y el simulador. Esta interfaz permite introducir los comandos y recibir los resultados.

### a) Ejecución en el Cliente

Este tipo de arquitecturas en las que el procesamiento se realiza en el cliente, permite a los estudiantes utilizar los simuladores de forma interactiva evitando la latencia de red asociada a la solución basada en ejecución en el servidor. La tecnología Java se adapta muy bien a este tipo de sistemas, ya que permite la entrega de software a través de Internet en forma de applets que permiten al estudiante un nivel de interactividad igual al que tendría con una aplicación local. Asimismo, Java puede ser utilizado para proporcionar a los estudiantes herramientas de colaboración integradas en los propios laboratorios virtuales, así como software de control para realizar el seguimiento y monitorización de las acciones realizadas por los estudiantes.



**Figura 1.2 Laboratorio virtual de procesado de señal**

Un ejemplo de esta aproximación es el desarrollado por la Universidad Estatal de Arizona, EE.UU., que ha construido un laboratorio virtual de procesado de señal digital (DSP). Este laboratorio está organizado en torno a un simulador Java de un editor DSP

gráfico denominado **J-DSP**, Java Digital Signal Processing, [JDS00]. En este laboratorio, los estudiantes de un curso de enseñanza superior sobre procesamiento de señal, utilizan el editor J-DSP para realizar simulaciones de algoritmos de análisis espectral, cálculo de estadísticos y diseño de filtros, entre otros [SU00].

El elemento fundamental del laboratorio es el simulador J-DSP, al cual acceden los estudiantes mediante un navegador Web tradicional con soporte Java, ver Figura 1.2. Entorno al simulador, se han definido una serie de prácticas de laboratorio publicadas mediante documentos HTML que incluyen formularios que los estudiantes deben completar para ser evaluados. Las prácticas consisten en los enunciados de los ejercicios que los estudiantes deben resolver mediante el simulador y un conjunto de cuadros de diálogo donde deben responder a una serie de preguntas, introducir comentarios y observaciones sobre los conceptos aprendidos, además de proporcionar los nombres de los ficheros generados con el editor J-DSP durante las prácticas.

**CROS**, es otro ejemplo de sistemas en los que el procesamiento se realiza en el cliente. CROS, desarrollado en la Universidad de Pisa, Italia, ofrece un simulador de un osciloscopio de rayos catódicos [AF01].

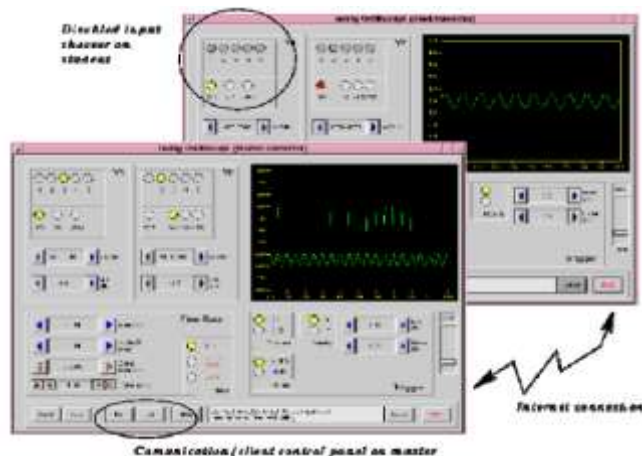


Figura 1.3 Ejemplo de una sesión cliente/servidor en el sistema CROS

El simulador permite a los estudiantes conocer la forma en la cual funciona un osciloscopio real. Todos los parámetros relevantes (valores para los umbrales, amplificaciones, histéresis, parámetros relacionados con el tiempo) se controlan desde la interfaz gráfica a la que accede el estudiante, y éste puede ajustarlos de forma intuitiva obteniendo una respuesta de forma interactiva, ver figura 1.3.

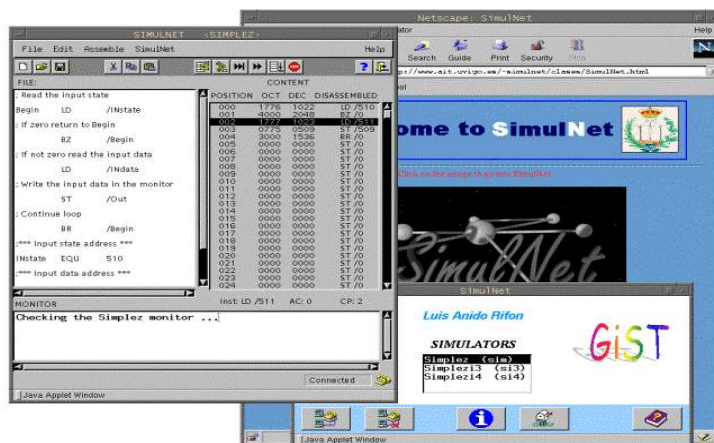
CROS puede ser utilizado como aplicación aislada sobre una estación de trabajo con Linux. También es posible que funcione en modo maestro/esclavo, estableciendo una conexión de red entre el estudiante y el profesor, de forma que el simulador de éste actúa como maestro y el del estudiante como esclavo. El profesor selecciona las entradas y configuración para exponer el problema que deben resolver los estudiantes. La configuración del simulador maestro se transfiere al cliente y el estudiante debe realizar los ajustes necesarios para encontrar la asíncrona, el profesor puede conocer el estado del simulador que actúa como esclavo y de este modo conocer el progreso del estudiante. Tanto el estudiante como el profesor se pueden comunicar mediante el envío de mensajes, de forma que el profesor pueda ayudar y guiar al estudiante hacia la solución correcta.

**SimulNet** es un sistema de Entrenamiento distribuido basado en computador, soportado en Web y CD-ROM. Al contrario de otros sistemas de entrenamiento basado en computador CBT (Computer Based Training) diferentes, cuyo objetivo es lograr un aula virtual, SimulNet proporciona un laboratorio virtual para poner el conocimiento teórico en práctica [AF01]. Esto se hace, de un lado, entregando el software necesario a través de Internet de tal forma que pueda ejecutarse en cualquier computador y, por otro lado, distribuyendo el laboratorio en el formato de CD-ROM. Estas aplicaciones distribuidas son simuladores de herramientas que pueden encontrarse en un laboratorio convencional.

Una vez el simulador descargado desde Internet, puede ser corrido interactivamente, como una aplicación local o la versión de CD-ROM. No hay ningún costo de red que pudiera reducir la interacción de usuarios con el sistema. A los estudiantes se les proporcionan un acceso remoto, y el simulador interactivo de una herramienta del laboratorio real. En algunos casos, esto sería bastante para el entrenamiento práctico. En el peor de los casos, la simulación puede usarse como un paso anterior antes de ir al laboratorio real. En tal caso, los estudiantes se familiarizan con las herramientas del laboratorio real a través de la simulación [AF03].

Es completamente esencial impedir que los estudiantes se sientan solos mientras se encuentran usando los sistemas de aprendizaje a distancia. Para evitar esto, SimulNet proporciona varias herramientas de comunicación, en modo síncrono y asíncrono. De esta manera los estudiantes y los maestros pueden conseguir estar en contacto superando las distancias geográficas.

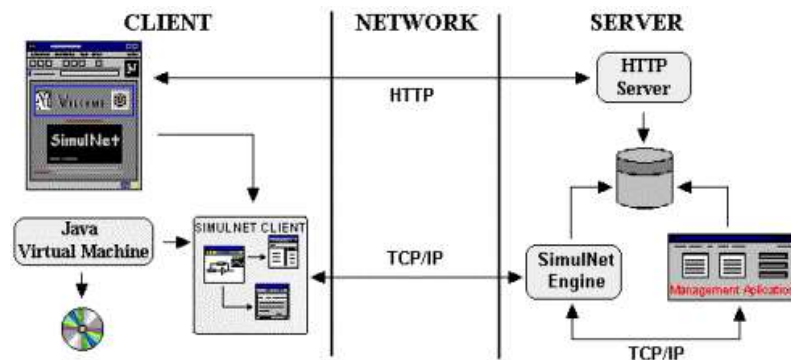
Los instructores de SimulNet están dotados de una herramienta de monitoreo y supervisión que les permite saber lo que sus estudiantes están haciendo en cualquier momento. De esta manera, ellos pueden usar esta información para organizar el material de aprendizaje y la metodología pedagógica que usarán para adaptarla a la experiencia y características de sus estudiantes. Al mismo tiempo, los docentes pueden usar esta herramienta para descubrir cuando un estudiante está en dificultades.



**Figura 1.4 SimulNet on the World Wide Web**

La implementación de SimulNet está basada en tecnologías de Internet disponibles normalmente, especialmente en Java, ver figura 1.4. SimulNet provee un cliente WWW (basado en Applets Java) y un cliente stand alone (basado en aplicaciones Java) distribuido mediante CD ROM. Los simuladores son interactivos. El uso de la red es transparente al usuario. Los simuladores corren aparentemente sobre el propio computador del estudiante. Esto es una característica esencial para proveer entrenamientos de la misma realidad como si ellos estuvieran en el laboratorio usando las herramientas de entrenamiento real [AF01].

El sistema está compuesto por varias aplicaciones Java 100% puras con otras para implementar cada facilidad SimulNet. La comunicación es lograda usando los protocolos de la capa de transporte de Internet TCP/IP. También se toma ventaja de los elementos WWW existentes: servicio http, HTML, WWW browser y servidor. La interacción de los elementos del sistema se muestra en la siguiente figura.



**Figura 1.5 Arquitectura Distribuida de Simulnet**

Los usuarios pueden acceder a SimulNet usando un browser WWW, ver figura 1.5, el cual presenta un documento HTML suministrado por el servidor http en el lado del servidor. Este documento contiene un applet Java embebido que inicia y detiene cualquier otra aplicación Java dada desde el lado del servidor a través de Internet. De esta forma, ningún software adicional es requerido aparte del browser en si mismo.

#### **b) Ejecución en el Servidor**

La otra posibilidad para laboratorios virtuales basados en simuladores consiste en su ejecución en el servidor. Los estudiantes envían los comandos al simulador a través de Internet. Éste, después de procesar el comando, envía las respuestas de vuelta a cada cliente. Obviamente, en este tipo de sistemas aparece un retardo de red que no existe en aquellos en los que el procesamiento se realiza en el cliente. No obstante, éste método es el adecuado en aquellos casos en los que los recursos necesarios para el simulador no puedan ser satisfechos por los ordenadores personales de los estudiantes.

Actualmente es posible encontrar varios sistemas que siguen esta filosofía. WebMath, desarrollado en la Universidad de Illinois en Urbana-Champaign, EEUU, proporciona un entorno de simulación para aprendizaje on-line asíncrono sobre antenas [Mat]. El

sistema permite la ejecución on-line de comandos con respuesta inmediata. Mathematica es el software comercial que eligieron los desarrolladores de WebMath para proporcionar animaciones y modelado de realidad virtual, y para desarrollar los paquetes de WebMath que muestran las propiedades fundamentales de los tipos básicos de antenas. La interfaz del estudiante está basada en applets Java y aplicaciones que permiten acceso remoto al simulador. Un CGI se encarga de recibir los datos introducidos desde el applet o la aplicación y de devolver los resultados.

La tecnología CGI también ha sido utilizada en Ibrahim, en la Universidad de Ginebra en Suiza. Ibrahim permite a los estudiantes no sólo tener acceso on-line a los contenidos de los cursos de Estructuras de Datos, sino también simular los algoritmos que se describen en los contenidos [AF01]. Los estudiantes pueden realizar la simulación en el servidor e interactuar con la misma introduciendo puntos de ruptura, breakpoints, en el código, accediendo al valor de variables, ejecutando paso a paso el código, etc. La interfaz con los estudiantes consiste en páginas HTML mostradas en sus navegadores.

La tecnología CGI fue utilizada en Ibrahim para monitorizar la ejecución de los programas que implementan la simulación. Dado que HTTP es un protocolo sin estado, las peticiones recibidas por el servidor Web se envían a un CGI que se encarga de analizar cada petición y de decidir si crea un nuevo proceso o le envía la petición a uno de los procesos ya existentes en el servidor.

### **c) Procesamiento Mixto**

Existen plataformas educativas que proporcionan simulaciones con procesamiento mixto, es decir, parte en el cliente y parte en el servidor. Un ejemplo de este tipo de sistemas es SPU [AF01], desarrollado en la Universidad Pacific de Seattle, EEUU, que proporciona un simulador de mercado bursátil.



Un reto en la enseñanza de microeconomía es que la idea de oferta/demanda tenga un significado real para los estudiantes. Una solución para ello es proporcionar a éstos un mercado virtual donde compradores y vendedores intenten llegar a acuerdos sobre los precios, y esto es lo que proporciona SPU. La estructura general del sistema de mercado ofrecido por SPU se detalla a continuación:

1. El estudiante observa los datos de mercado. Éstos consisten en información de dos tipos: ofertas actuales y negociaciones completadas.
2. El estudiante realiza una oferta. Cada oferta contiene la siguiente información: el tipo de la oferta (de compra o de venta), el precio y la cantidad de acciones que ofrece.
3. El sistema procesa la oferta y devuelve los resultados de la misma. Básicamente hace una correspondencia entre ofertas de compra y venta.

SPU utiliza diferentes tecnologías Web para realizar la simulación. Por un lado procesamiento en el servidor, donde residen los datos del mercado. Los estudiantes piden estos datos al servidor, y éste se los devuelve al navegador Web o mediante correo electrónico. La oferta del estudiante se transmite al servidor también desde el navegador o a través de correo electrónico. El servidor procesa las ofertas, busca correspondencias entre compras y ventas, y las devuelve al estudiante como transacciones completadas. La implementación del servidor está basada en las tecnologías CGI y ASP [AF01].

Por otro lado existe procesamiento en el cliente. Los estudiantes observan los datos del mercado en sus ordenadores. Las ofertas se transmiten desde el cliente del estudiante al servidor. El servidor envía la oferta a todos los clientes conectados en ese momento. Los clientes reciben información de las ofertas existentes desde el servidor y gestionan el procesamiento y visualización de los resultados. Se utilizan applets Java como base para la implementación.

No existe una regla general para establecer cuál de las tendencias presentadas anteriormente es la mejor. La solución depende del área en la que se encuadre el laboratorio virtual, de los recursos computacionales disponibles tanto en los clientes como en el servidor y el ancho de banda disponible en la red.

En conclusión, históricamente, los nuevos sistemas de aprendizaje a través de Internet son una evolución de los sistemas de educación a distancia tradicionales, a los que se le han añadido numerosas funcionalidades mediante el empleo de las nuevas tecnologías de la información. Durante los últimos años han aparecido cientos de estos sistemas, con ellos se quiere proporcionar aplicaciones que permitan mantener entornos de aprendizaje virtuales útiles y eficaces para todos los actores involucrados.

El análisis realizado ha establecido una separación entre los sistemas que entregan recursos educativos orientados al aprendizaje teórico (principalmente expositivos) y aquellos orientados al aprendizaje práctico (principalmente activos). El desarrollo de los segundos se ha realizado más tarde que los primeros debido a una superior dificultad para su desarrollo y a unos requisitos tecnológicos mayores. Precisamente por ello, el número de sistemas que implementan el paradigma del laboratorio virtual en entornos de aprendizaje electrónico a distancia es muy inferior al de aquellos que ofrecen entornos de aulas virtuales.

En este orden de ideas y pensando en el actual contexto de la sociedad de América latina, específicamente en las nuevas demandas educativas, la aplicación del paradigma learning by doing al proyecto E-LANE, se constituye en un valioso aporte, por cuanto permitirá generar un gran valor agregado en materia de formación, mediante un contenido que se caracteriza por poseer una alta interactividad, buscando así, mejorar la calidad y los resultados del e-learning. Para la Universidad del Cauca y en el actual proceso de descentralización que inicia, la aplicación del paradigma learning by doing representa una aproximación inicial a las nuevas demandas que en materia de

educación le hace la región.

### **1.3 PROYECTO E-LANE ( EUROPEAN AND LATIN AMERICAN NEW EDUCATION)**

La Universidad del Cauca en su afán de no rezagarse en cuanto la aplicación de las Tecnologías de la información y la comunicación (TICs) a los procesos formativos, con los consiguientes beneficios que la educación en ambientes virtuales representa, es un participante activo en el proyecto E-LANE (European and Latin-American New Education), el cual propone impactar en los procesos formativos en distintos escenarios de Latinoamérica, a través de la integración de aplicaciones sólidas usadas en el contexto de aprendizaje para la construcción de una plataforma de aprendizaje de programación abierta, el diseño de metodologías de aprendizaje innovativo orientadas hacia dicha plataforma, y la integración de cursos de algunas prestigiosas instituciones educativas en Europa y América Latina con el objetivo de proveer el acceso a material educativo de bajo costo [E-lane].

Dentro de los objetivos específicos que persigue este proyecto se encuentran:

**Plataforma de programación abierta para tele-enseñanza.** Integración del estado del arte de las Tecnologías de la Información relacionadas con la tele-educación, para obtener una plataforma de software libre, con mínimos requisitos de equipamiento, que suministre la funcionalidad esencial para orientar cursos en diferentes disciplinas y escenarios.

**Metodología Innovadora.** De igual importancia que la plataforma de programación es una metodología innovadora que saque provecho de los nuevos recursos dispuestos por la tecnología. Más concretamente, tanto la manera de orientar los cursos como la forma de preparar el material, deben tener en consideración el uso de un entorno de tele-educación.

**Contenidos de los Cursos.** Se producirá un número determinado de cursos para ser orientados con las herramientas y metodologías propuestas.

**Medición de Impacto y Refinamiento.** El impacto social es de la mayor importancia. El objetivo de la propuesta es maximizar tal impacto; por consiguiente, es esencial una etapa de evaluación planeada cuidadosamente.

**Difusión y Sostenibilidad.** Este objetivo está orientado hacia la ampliación del impacto de la plataforma, no sólo en los estudiantes que reciben educación con ella, sino también en toda la comunidad educativa y científica.

El proyecto E-LANE propone la integración de diversas herramientas telemáticas, combinadas con el cumplimiento de estándares internacionalmente utilizados, para dar una plataforma de aprendizaje de código abierto capaz de brindar servicios adecuados de difusión de e-learning, a un determinado número de usuarios: dotLRN. En otras palabras dotLRN es su sistema de gestión de aprendizaje (LMS).

Todos los Sistemas de Gestión de Aprendizaje no fueron creados igual. Como la mayoría de los sistemas software, cada LMS está basado sobre un punto de vista específico de cómo hacer cumplir mejor los requerimientos particulares de usuario. Estos requerimientos varían de sistema a sistema. Por ejemplo, unos se enfocan sobre la gestión del contenido y las herramientas de evaluación, haciendo del contenido, el centro del diseño. Otros dan a los sistemas workflow más énfasis. Esta especialización ha creado LMS con fortalezas en diferentes áreas. En este orden de ideas y con el ánimo de establecer un punto de referencia para la aplicación del paradigma learning by doing al proyecto E-LANE, es necesario hacer una caracterización funcional y conceptual de un LMS, por cuanto permite entender el contexto y el alcance del proyecto, objeto de estudio, para finalmente centrar la atención de manera específica en dotLRN y en los laboratorios virtuales como aplicaciones e-learning.

## 1.4 SISTEMAS DE GESTION DE APRENDIZAJE (LMS)

Sistema de Gestión de Aprendizaje es un término que es usado para un amplio rango de sistemas que organizan y proveen acceso a servicios de aprendizaje online para estudiantes, profesores, y administrativos. Estos servicios usualmente incluyen control de acceso, provisión de contenido learning, herramientas de comunicación, y administración de grupos de usuarios. A menudo un LMS es referido como una plataforma de aprendizaje. Kaplan-leiserson define un LMS como un software que automatiza la administración de eventos de entrenamiento. El LMS registra usuarios, registra datos de los estudiantes y provee reportes para gestión. Un LMS está típicamente diseñado para manejar cursos y se enfoca en la gestión. Clark Aldrich establece como objetivos de un LMS, dar el contenido correcto a la persona correcta en el momento correcto, así como registrar y reportar el evento [AC02].

Durante los últimos años han aparecido varios de estos sistemas, con ellos se quiere proporcionar aplicaciones que permitan mantener entornos de aprendizaje virtuales útiles y eficaces para todos los actores involucrados. Desde el punto de vista de la arquitectura, un LMS debería tener las siguientes características:

**Abierta:** El objetivo es crear aplicaciones e-learning interoperables y conectables entre si de forma sencilla, es decir, que herramientas comerciales de fabricantes distintos puedan ensamblarse en único sistema global.

**Escalable:** Independiente del tamaño inicial con que se conciba el sistema, la arquitectura debe estar definida da tal forma que permita su crecimiento. Por ejemplo, al ir aumentando el repositorio de objetos educativos las aplicaciones encargadas de gestionarlos deben tener capacidad suficiente para no sobrecargarse.

**Global:** Permitir la diversidad lingüística y cultural. Este es uno de los objetivos con mayor dificultad, puesto que la gran mayoría de las aplicaciones están destinadas para una audiencia anglosajona.

**Integrada:** No solo entre los componentes del propio sistema, si no entre otras aplicaciones no directamente relacionadas con el aprendizaje. El objetivo es conseguir interoperabilidad entre ellas.

**Flexible:** Es importante la capacidad de implementar nuevas soluciones sin tener que efectuar grandes cambios en la arquitectura del sistema.

Por otro lado, la arquitectura está fuertemente condicionada por los roles definidos para los actores que interactúan con el sistema. En este sentido se identifican tres categorías básicas entre los usuarios de un entorno e-learning: consumidores de e-learning, a esta categoría pertenecen estudiantes y profesores; proveedores de e-learning, a esta categoría pertenecen los diseñadores de contenidos educativos; coordinadores, a esta categoría pertenecen los administradores del sistema.

Aunque las funcionalidades de un sistema de enseñanza pueden ser muchas y muy variadas, se pueden agrupar en tres sistemas independientes y bien diferenciados [GP04].

### **Sistema de gestión de contenidos**

Este grupo engloba todas aquellas tareas relacionadas con la creación, recuperación, ensamblado, localización y almacenamiento de contenidos y recursos educativos.

### **Sistema de distribución de contenidos**

Su función es determinar la forma en que se van a presentar los contenidos al usuario, para ello es recomendable disponer de las herramientas adecuadas para llevar a cabo funciones como: presentación de contenidos y gestión de la distribución.

### **Sistema de gestión del aprendizaje**

Esta área esta asociada el rol de los consumidores de e-learning. El término LMS, por lo general, hace referencia a un tipo de sistema diseñado para realizar el seguimiento de la interacción con los contenidos educativos y realizar la gestión de los mecanismos de interacción con el sistema, incluyendo la navegación, la selección de las ofertas educativas y la conexión con el sistema de distribución de contenidos.

El objetivo primario de un LMS es proporcionar y administrar los diferentes recursos de un sistema de enseñanza a los estudiantes, siguiendo su progreso y su rendimiento [GP04]. En particular, las características principales de un LMS, que lo diferencian de las herramientas tradicionales de enseñanza son: personalización, búsqueda y navegación en catálogos educativos, registro, seguimiento de la evolución del estudiante, mecanismos de valoración y evaluación.

Una vez fijadas las características generales de un LMS se hace necesario centrar la atención de manera específica en dotLRN como el LMS del framework OpenACS, específicamente en sus funcionalidades generales que le darán valor agregado a un laboratorio virtual como aplicación de aprendizaje, así como también en aquellos paquetes que permitirán la integración de estos, y así poder justificar la propuesta descrita en el capítulo dos y su posterior validación en el capítulo tres.

## **1.5 PLATAFORMA DOTLRN**

Las diversas plataformas de tele-educación proveen principalmente soporte para tareas administrativas genéricas, como también algunas herramientas básicas para soportar las actividades de enseñanza y aprendizaje. De forma paralela, en años pasados ha habido una actividad muy significativa para definir diferentes estándares que permitan el intercambio de material de aprendizaje entre diferentes plataformas [CE03].

En esta situación, el proyecto E-LANE propone la integración de diversas herramientas

telemáticas, combinadas con el cumplimiento de estándares internacionalmente utilizados, para dar una plataforma de aprendizaje de código abierto capaz de brindar servicios adecuados de difusión de e-learning, a un determinado número de usuarios.

La plataforma tiene una arquitectura cliente servidor e impone unos requerimientos mínimos del lado del cliente [Open]. Los diferentes subsistemas bajo los cuales la plataforma esta basada son: Base de datos PostgreSQL, AOLserver, OpenACS, y dotLRN entre otros [CG04].

**PostgreSQL** es un sistema de gestión de base de datos relacional (RBDMS) que soporta completamente SQL (*Structured Query Language*). SQL es un lenguaje estandar internacional, comúnmente aceptado por los fabricantes de generadores de bases de datos tales como Oracle. PostgreSQL maneja la mayoría de instrucciones SQL avanzadas , y es consecuentemente reconocido por los expertos, como el más avanzado sistema de gestión de base de datos relacional (RBDMS) libre [Post].

**Servidor AOLserver:** AOLserver puede ser usado para gran escala, sitios web dinámicos, debido a la tecnología muy especial que utiliza, el servidor es multithread, lo cual asegura que es capaz de atender muchas peticiones simultáneamente. Por medio del lenguaje de programación TCL embebido en su corazón, el servidor AOLserver logra resolver muchos de los problemas asociados con CGI , el mecanismo más común para ejecutar scripts del lado del servidor.

**Gruopware OpenACS:** OpenACS (Sistema Comunitario de Arquitectura Abierta) es una colección de aplicaciones desarrollada gradualmente, diseñada para soportar comunidades web altamente dinámicas y sitios web interactivos [Open]. Este consiste de una colección extensiva de módulos; la mayoría de ellos integrados a través de una colección única de tablas de bases de datos que refieren a otra en una base de datos relacional [rubic].

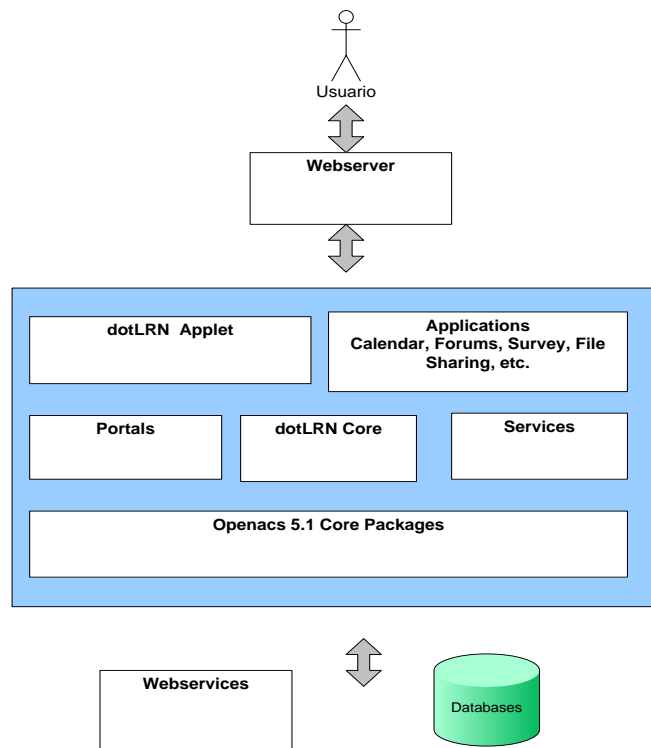


El modelo de datos estándar de OpenACS es muy práctico por varias razones. Para el usuario, porque permite seguir la personalización extensiva de características a través de todas las aplicaciones de un sitio particular. Para los administradores porque OpenACS provee con un único sistema template, un paquete gestor y sistemas de permisos para todas las aplicaciones.

DotLRN es un paquete integrado de aplicaciones de código abierto diseñado para el e-learning; se compone de un conjunto de módulos basados en la web que soportan gestión de cursos y comunidades online; y un portal framework soportando comunidades involucradas en el aprendizaje y la investigación [Open].

La arquitectura almacena toda la información en una base de datos relacional. El sistema permite utilizar Oracle, así como también Postgres, una base de datos distribuida como software libre. Los usuarios, o incluso administradores del sistema tienen una interface web que les permite crear departamentos y escuelas dentro de los cuales se distribuyen los cursos. Cada uno de estos cursos tiene una página creada automáticamente (“portal”) dentro de la cual aparecen todas las funcionalidades que el profesor considere importante para un curso particular. La apariencia de este portal puede ser adaptada por el profesor sin necesidad de programar. Cada usuario del sistema (Profesor, Estudiante, Ayudante, Administrador, etc) tiene una página personal donde la información de todos los cursos y comunidades a las que pertenece es compilada y presentada en forma uniforme. Estas funcionalidades se traducen en la activación de los módulos que se describen al final.

La arquitectura del framework OpenACS se muestra en la figura 1.6. El diagrama tiene cinco componentes principales que a continuación se nombran: usuario, webserver, framework, database y servicios [CG04].



**Figura 1.6 Arquitecturas OpenACS y dotLRN**

La plataforma dotLRN presenta un número de aplicaciones colaborativas tales como calendarios, foros, compartir archivos, etc. sumadas a las que el framework de OpenACS provee, (por ejemplo e-commerce, content management, survey, workflow, webmail, etc.). Muchas de estas funcionalidades son similares a las encontradas en otros LMS tales como WebCT, Blackboard y Moodle.

### 1.5.1 DotLRN user portal

La arquitectura está compuesta por los paquetes acs-core que proveen una API para la funcionalidad común, además de otros servicios y paquetes [CG04]. Puesto que

dotLRN está construido sobre el framework OpenACS, hereda todas las funcionalidades OpenACS, la modularidad y ventajas por medio del uso de interfaces que interactúan con los servicios y funcionalidades de OpenACS. Como ejemplo, dotLRN utiliza la aplicación calendario OpenACS simplemente personalizando algunas de sus interfaces. Varias instancias calendario pueden ser atadas o unidas a otras en un portal individual, desplegando en una única página todos los eventos de cada curso en que el estudiante está inscrito. Cada vista de un paquete es llamado un "portlet".

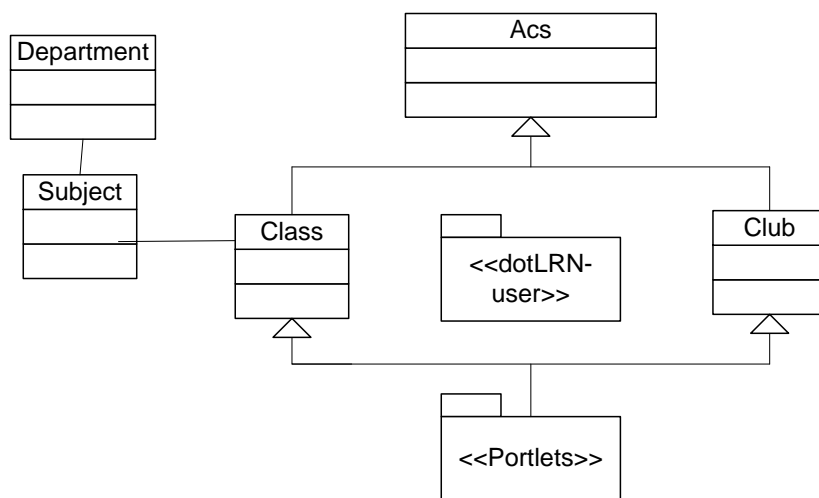


**Figura 1.7 dotLRN User Portal**

La figura 1.7 muestra un Portal de Usuario. El sistema adapta el portal a cada usuario recogiendo la información sobre los cursos y comunidades en los que el usuario está matriculado o pertenece, y qué aplicaciones de estos cursos y comunidades están usando. Cada una de estas aplicaciones generará contenidos apropiados mostrados en un portlet.

## 1.5.2. DotLRN communities

Las comunidades sirven para compartir un espacio de trabajo, varios integrantes de la comunidad pueden aprender a través de los cursos que cogen de la comunidad y aportar a esta sus estudios. En una compañía, las comunidades podrían representar grupos de interés especiales, proyectos o simplemente recolecciones sociales. En la universidad, clubes y sociedades son comunes. Se apoyan ambos tipos de actividades en dotLRN. Esto requiere scoping muy claro de cada aplicación para que cada course/community pueda tener sus propias aplicaciones fijas, plantillas y permisos. dotLRN usa el concepto “subsites” que separa los contextos entre los diferentes tipos de comunidades (“las clases” y los “clubes”, para nosotros asignaturas y comunidades). Cada caso del subsite (“dotLRN-community”) puede tener su propia estructura de permisos.



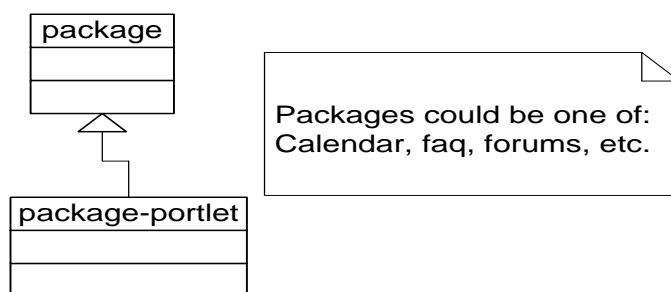
**Figura 1.8 Diagrama de Clases para dotLRN**

La figura 1.8 muestra en un diagrama de clases cómo las comunidades de dotLRN pueden ser clases o clubes, ambos tipos proporcionan juegos diferentes de

herramientas de colaboración, los roles para personas y plantillas con valor predeterminado. En ambos casos, estas comunidades tienen su propia instancia de calendarios, foros, las noticias, los grupos, el archivo compartido, etc. La figura 1.8 también muestra esta estructura y cómo las clases relacionan el departamento con la clase subject. “Los departamentos” describen la estructura orgánica normal dentro de la institución, “el Subject” contiene información que repite en los diferentes casos de un “Class”. “Class” especifica personas y sus papeles, la salida y la fecha de acabado, así como otros atributos.

### 1.5.3 Portlets

DotLRN usa la arquitectura portal OpenACS para generar la interface portal de usuario [CG04]. Los portlets dotLRN implementan la interfaz apropiada entre las aplicaciones OpenACS y el portal dotLRN. La figura 1.9 muestra un diagrama de clase para el diseño de un portlet.

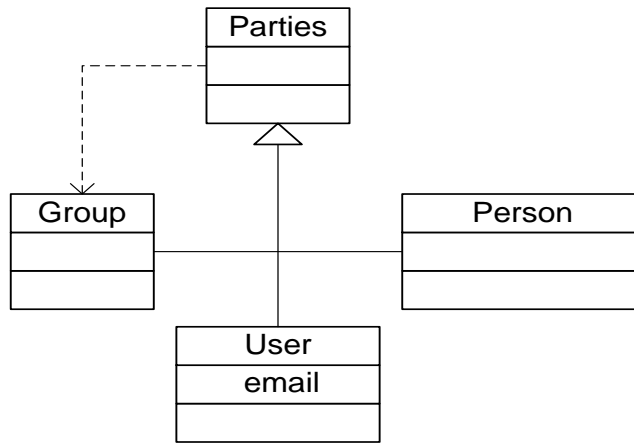


**Figura 1.9 Paquete de clases portlets dotLRN**

### 1.5.4 DotLRN users

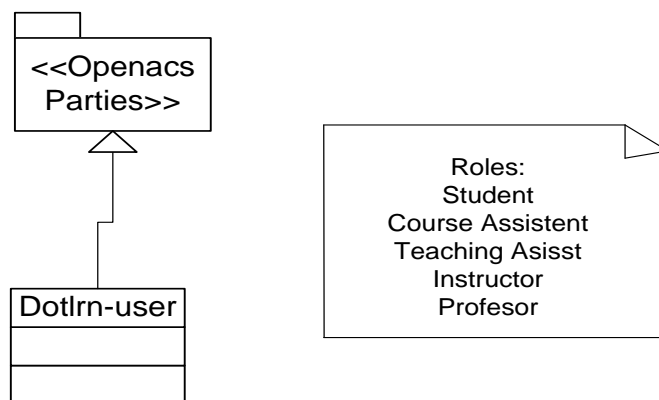
OpenACS puede soportar jerarquías de usuarios complejas mediante la definición de “parties” que pueden ser: “users” identificables por un email diferente, o una “person”

(persona) con el primer y último nombre distintivo; ó un grupo que puede ser uno o más de los de abajo [Open]. Estas relaciones se muestran en la figura 1.10.



**Figura 1.10 Paquete parties OpenACS**

Con esta arquitectura los usuarios dotLRN pueden mantener más de un role particular [CG04]. Por ejemplo, un usuario podría ser asistente de enseñanza en una clase y un estudiante en otra. Por consiguiente el papel de un usuario en una clase podría determinar lo que le está permitido hacer. La jerarquía de usuario está desplegada en la figura 1.11.



**Figura 1.11 Paquete de Usuario dotLRN**

### **1.5.5 Módulos de dotLRN:**

#### **a) DotLRN**

La funcionalidad de cualquier paquete de prueba de dotLRN es dependiente de la funcionalidad de OpenACS y es luego exportada al corazón de dotLRN. Un paquete dotLRN es un applet que debe proveer un API simple bajo el mecanismo acs-service-contract. Esta API permite al corazón de dotLRN despachar llamadas a cada applet cuando ciertos eventos suceden. Luego, un applet dotLRN debe ser capaz de responder a los siguientes eventos: 1) el applet es añadido a una comunidad, 2) el applet es removido desde una comunidad, 3) un usuario es añadido a una comunidad para la cual el applet en cuestión está habilitado; 4) un usuario es removido desde una comunidad para la cual el applet en cuestión está habilitado.

Los eventos adicionales pueden ser añadidos en el futuro y así lograr una función applet generalizada. La mayoría de los applets dotLRN buscan ofrecer una interfaz a sus usuarios. Para lograr esto, el corazón dotLRN utiliza la nueva arquitectura portal de OpenACS. Un applet dotLRN puede simplemente añadirse así mismo a las páginas portal apropiadas por medio de un portlet de la "New Portal Architecture". Un paquete dotLRN está usualmente compuesto de tres paquetes OpenACS:

- 1) Un paquete de funcionalidad corazón (por ejemplo calendar), independiente de la funcionalidad dotLRN.
- 2) Un paquete portlet (por ejemplo calendar-portlet), cuya interfaz para la New Portal Architecture es independiente de la funcionalidad dotLRN.
- 3) Un applet dotLRN (por ejemplo dotLRN-calendar) dependiente de la funcionalidad dotLRN y básicamente el adhesivo entre el corazón dotLRN y la funcionalidad de este

nuevo applet.

## **b) DotLRN-calendar**

Un calendario es un agregado de eventos los cuales caen dentro de un periodo de tiempo dado, por ejemplo, un día particular, semana, o mes [CG04]. La aplicación calendario provee a los usuarios de una herramienta que permite añadir, ver, editar y organizar estos eventos tanto en los niveles personal y grupo/compañeros.

El paquete calendario está construido sobre la cima del paquete de servicio ACS Event y alcanza los siguientes tres niveles de funciones:

**Gestión de eventos:** Cubre los aspectos de la aplicación que permite a eventos, tales como añadir, editar, ver, eventos borrados, eventos de fijado recurrente.

**Vistas calendario:** Cubre aquellos aspectos de la aplicación que permiten el despliegue de eventos calendario para un periodo de tiempo particular.

**Navegación:** Cubre aquellos aspectos de la aplicación que permiten formas en las cuales el usuario final puede cambiar el periodo de tiempo normalmente desplegado y los eventos asociados.

## **c) DotLRN-fs: File storage module**

La aplicación file-storage permite a los individuos colocar sus archivos sobre un sitio Web accesible y compartirlos con otros miembros de esa comunidad web o con el publico en general. La aplicación file-storage permite a los usuarios habilitarlos para mantener colaborativamente un conjunto de documentos. Específicamente, los usuarios



pueden :

- 1) organizar archivos en un directorio, en una estructura de directorio jerárquico
- 2) subir usando formatos web, usando la característica del navegador web para subir archivos.
- 3) recuperar versiones anteriores de un archivo.

Los objetos principales de File Storage son “folders” y “archivos”. Un “folder” es análogo a un subdirectorio en el sistema de archivos UnixWindows-world. Los objetos Fólder son almacenados como fólдерes del repositorio de contenido

#### **d) DotLRN-forums**

El módulo foros permite a los usuarios dotLRN interactuar con otros usuarios del mismo curso o personas del mismo club para hablar o discutir tópicos o consultas. Los foros permiten una discusión libre para todos (moderada en un cierto nivel por el administrador del sistema). Esto permite a los usuarios iniciar su propio thread, enviar mensajes a sus amigos como attachments y buscar un tópico requerido en el archivo de mensajes del foro

#### **e) DotLRN-faqs**

Este módulo permite al administrador y los profesores enviar FAQ's sobre el curso o el portal comunitario. Esto permite a los miembros del curso o la comunidad leer información a cerca de varias cosas en primer lugar. Estas FAQs pueden servir como punteros a URLs, guías de instalación, guías de usuario, etc.

#### **f) DotLRN-news**

Este módulo puede ser usado en varias formas, ya sea en forma de diseminación o

advertencia. Los ítems noticias son tales que pueden dejar de ser noticias después de un tiempo. Cuando esto sucede, estos ítems deben desaparecer automáticamente dentro de los archivos de noticias sin la intervención del administrador.

El cuerpo de las noticias puede tener un formato MIME de “text/plain” o “text/html”. Usando HTML, el publicador puede enlazar archivos de imágenes, video, o audio dentro del cuerpo de la publicación desde otros sitios o desde el módulo local de guardar archivos.

#### **g) DotLRN-homework**

El paquete Homework dotLRN provee un dropbox para que los estudiantes puedan subir archivos homework, así como también para profesores y asistentes suban sus comentarios y correcciones de tales archivos. La interfaz de usuario es muy similar a la proveída por el paquete estándar de guardar archivos. El paquete homework está diseñado para ser montado bajo cada instancia de clase dotLRN. Los administradores pueden hacer subfolderes en fólderes dropbox de la clase homework. Típicamente esto sería usado para crear un subfolder para cada tarea homework. Los administradores pueden pedir ser notificados mediante e-mail cuando un estudiante sube un nuevo archivo homework. Los estudiantes son notificados cuando un archivo de comentario o corrección es subido por un evaluador. Los estudiantes pueden solamente acceder a los archivos homework que han subido y a los archivos asociados con comentarios y correcciones.

Finalmente es necesario resaltar que numerosas universidades han optado por dotLRN como plataforma para administrar sus cursos. Debido a su historia, Sloan MIT fue el motor de este proyecto durante sus dos primeros años. Hoy día, otras universidades como Heidelberg en Alemania, y Viena en Austria están invirtiendo considerables recursos en esta dirección. Heidelberg y Viena tienen aproximadamente 30.000 estudiantes cada una, muchos más que Sloan; otras instituciones incluyen Cambridge

en Inglaterra, Birzeit en Palestina, Berkeley en EEUU.

Todas estas universidades tienen mucho en común y algunas diferencias. Algunas son extremadamente grandes y antiguas, Heidelberg es una de las más antiguas en Europa. Otras son pequeñas y nuevas, como la nueva Birzeit, única universidad en Palestina. Algunas instituciones, como Heidelberg, son universidades estatales y casi gratuitas. En forma similar a lo que sucede en muchas universidades argentinas, Heidelberg está buscando en Internet una solución que le permita mejorar la calidad de la enseñanza para los estudiantes numerosos de primer año. En otros casos como el de Berkeley College of Music, la universidad buscaba nuevas oportunidades de negocios, en particular, una infraestructura que les permitiese vender cursos, libros y otros productos on-line. Cambridge en el Reino Unido y la Universidad de Sydney lo usan para enseñar a una fracción pequeña de estudiantes (comparada con el total de estudiantes de la universidad), pero han encontrado en dotLRN una herramienta muy útil para mejorar la colaboración entre sus académicos. La Universidad de Sydney lo utiliza también en sus proyectos de investigación.

## **II. ARQUITECTURA PROPUESTA**

El presente capítulo tiene por objetivo describir la Aplicación del paradigma learning by doing al proyecto E-LANE mediante la implementación de laboratorios virtuales en la plataforma dotlrn de OpenACS; proyecto que se soporta teóricamente en el capítulo uno, correspondiente al estado del arte del paradigma learning by doing. El capítulo contiene la descripción de los requisitos esenciales, el modelo de referencia funcional, y la arquitectura propuesta para el mismo. Cada uno de estos ítems se presenta como un subproducto de acuerdo al Modelo para la Construcción de Soluciones.

### **2.1 MODELADO DE REQUISITOS ESCENCIALES**

En el desarrollo de un proyecto, el modelado de los requisitos esenciales se constituye en la definición de las funcionalidades básicas del sistema, en este orden de ideas la definición de los requisitos esenciales para la aplicación del paradigma “learning by doing” en el proyecto E-LANE se soporta teóricamente en las consideraciones hechas en el capítulo uno, cuyo análisis y estudio detallado fijaron el camino para el establecimiento de los requisitos y la arquitectura.

Inicialmente se establecieron las aproximaciones del paradigma learning by doing, en la cual se fijó una caracterización de los laboratorios virtuales más representativos que lo implementan; características que se describieron en términos de las funcionalidades, arquitecturas de referencia y la base tecnológica que ellos presentan en su conjunto.

Partiendo del hecho que las características fundamentales del e-learning [GP04], se basan en la combinación de tres factores: seguimiento, contenido y comunicación, las cuales se encuentran tanto en proporción variable como en función de la materia a tratar, es lógico pensar que éstas deben ser extendidas a los laboratorios virtuales para

complementarse en una sola unidad de alto valor pedagógico.

Es desde este punto de vista, donde la consideración de tales funcionalidades se constituyen en un requisito que justifique el modelado de los requisitos para la aplicación del paradigma learning by doing al proyecto E-LANE; mientras que la caracterización de las arquitecturas y la base tecnológica, en conjunto con la estudio y exploración de la plataforma OpenACS y su LCMS, así como el soporte teórico para un laboratorio virtual darán las bases para justificar la arquitectura propuesta y la implantación de la misma en la plataforma objeto de estudio.

Las funcionalidades que un laboratorio debe proveer deben estar enmarcadas dentro de las tres características fundamentales del e-learning, (seguimiento, contenido, comunicación), sin embargo ninguno de los laboratorios que se estableció como objeto de estudio bibliográfico, está integrado a un sistema de gestión de e-learning (LMS); lo cual limita toda las posibilidades que un laboratorio virtual puede llegar a proveer. Incluso la gran mayoría son distribuidos de manera aislada y fuera de comunidades e-learning.

Considerando que un salón virtual es aún un remanente del modelo en el cual el instructor “hace” y los estudiantes “ven”, un análisis de los laboratorios virtuales del capítulo uno muestra que los laboratorios serán una importante diferencia y su distribución se complementa a la instrucción síncrona que ocurre en un salón de clase virtual, resultando en una experiencia más efectiva para el estudiante [TR02].

Los salones virtuales de clase son una aproximación altamente efectiva y escalable para llegar a un mayor número de estudiantes. Pero aún los salones virtuales algunas veces tienen sus limitaciones, en términos de los mejores tipos de entrenamiento, comunicaciones escalables instructor-estudiante, procesos efectivos de acceso y gestión de objetivos de aprendizaje. Estos salones virtuales típicamente se soportan en un aprendizaje del estudiante por observación, en quien delegan toda la responsabilidad del mismo [TR02].

En este orden de ideas los laboratorios virtuales muestran un contenido que se caracteriza por poseer una alta interactividad, buscando mejorar la calidad y los resultados del e-learning. En estos laboratorios la interactividad es vista como una parte de un sistema donde los aprendices no son recipientes pasivos de información, sino que hacen parte del material que tiene un comportamiento activo frente a sus acciones. La interactividad resulta en un aprendizaje más sólido debido a que los aprendices pueden hipotetizar para probar su entendimiento, aprender mediante los errores y darle sentido a lo no significativo. La interactividad busca hacer del e-learning una experiencia personalizada que se construya basada en el estado de interacción del estudiante con los objetos.

En las teorías constructivistas, el diálogo entre los aprendices se considera fundamental tanto en la construcción de su propio entendimiento como en la validación del mismo por medio de la discusión con sus semejantes [TR02]; Siguiendo esta directriz y regresando al capítulo uno, sólo algunos laboratorios brindan capacidades de comunicación con el ánimo de proveer una valiosa experiencia en la cual los aprendices compartan más que palabras e imágenes.

En un entorno e-learning la comunicación integrada a los laboratorios debe hacer que la experiencia sea más rica que la educación tradicional [TR02]. El e-learning tradicional le permite a los aprendices acceder el contenido de acuerdo a su propia conveniencia, pero ellos aprenden solos, separados de otros, separados por la distancia y el tiempo. Tales estudiantes pueden dejar de lado el sentido de comunidad que la interacción con otros aprendices y tutores puede construir. La comunicación estudiante-estudiante es a menudo subestimada pero es críticamente importante en tareas colaborativas, que requieren habilidades de trabajo en equipo donde el dialogo debe ser primordial. En el e-learning, las comunidades están siendo desarrolladas a través de la utilización de técnicas asíncronas tales como bulletin boards y chat rooms.

Cuando un laboratorio virtual es agregado a un LMS, los aprendices y tutores necesitan comunicarse con más que palabras e imágenes [TR02]; los estudiantes y los tutores deben compartir las ventajas y oportunidades que el aprendizaje práctico ofrece, y así explorar las diferentes posibilidades y alternativas del aprendizaje en grupo.

En el e-learning, dependiendo del tipo de aplicación, el rol de cualquier tutor puede variar desde ser un instructor, guía, experto o facilitador. Los tutores pueden intervenir para corregir concepciones erradas, responder preguntas, motivar el entendimiento del estudiante o promover el diálogo. El acceso compartido a los laboratorios virtuales es importante para la comunicación en todos estos escenarios.

Así como incrementalmente, los cursos online están siendo distribuidos a través de sistemas de gestión de aprendizaje (LMS), los laboratorios virtuales a pesar de ser una aproximación e-learning nueva, no son la excepción, con lo cual se busca proveer un conjunto de herramientas para gestionar y distribuir todas sus capacidades. En este contexto facilitar la interoperabilidad y la integración de los laboratorios con estas herramientas en un LMS, puede distribuir beneficios sustanciales para la calidad del e-learning.

El capítulo uno también permitió establecer la caracterización de los componentes básicos que forman un laboratorio virtual al estar integrado en un LMS, los cuales se ubican en el campo de la Gestión y la Distribución. El primer componente se traduce en las capacidades que habilitan la gestión y la planeación de los laboratorios. Los componentes claves de gestión para un laboratorio virtual típicamente deben estar dirigidas a las siguientes capacidades de gestión: procesos para acceso y configuración, debe incluir reportes sólidos que ayuden a entender el uso de la aplicación, así como de otras fuentes; y en general todas las capacidades que le brinda el LMS.

El componente de la distribución involucra las capacidades que dan a los estudiantes

una experiencia segura, personalizada y altamente interactiva. La distribución de un laboratorio virtual involucra considerar los siguientes elementos: acceso seguro, captura de sesión, otras. y en general todas las capacidades que le brinda el LMS.

Los dos componente anteriores, son entregados a través del equivalente de una “solución stack” consistente de tres capas funcionales. Una capa de “aplicación”, en la cual se encuentran las aplicaciones consideradas como laboratorios, y corresponde a la capa por medio de la cual el usuario interactúa con el sistema. Esta capa se corresponde con otras aplicaciones e-learning que complementan la funcionalidad del laboratorio; en términos de comunicación a través de herramientas asíncronas o síncronas; en términos de seguimiento y evaluación, a través de herramientas assesment y herramientas de seguimiento y retroalimentación. Una capa “plataforma”, que puede ser entendida como una colección integrada de clases que busca generar servicios de valor añadido para la gestión de los laboratorios que habilitan las capas de aplicación. Por lo general esta capa no tiene una interacción directa con el usuario, pero representa un alto valor para los objetivos del sistema. En un nivel más técnico, un laboratorio virtual tiene que construir o buscar el valor agregado de determinados servicios en un LMS, tales como gestión y control inteligente, configuración, gestión del número de estudiantes, la duración de uso. Estos servicios son expuestos a administradores, instructores, estudiantes, permitiendo a ellos alcanzar sus trabajos, planear cursos y otras actividades e-learning. Finalmente la “infraestructura virtual”, que puede ser vista como la capa que se fundamenta en la tecnología software subyacente cuyo objetivo se traduce en habilitar la utilización de fuentes de bases de datos y las funcionalidades de los entornos de red. Corresponde a la capa inferior del stack, y representa un gran valor agregado en términos de comunicaciones y fuentes de datos.

La caracterización de estos requisitos, debe conducir a la generación de la arquitectura necesaria para la Aplicación del paradigma learning by doing mediante la construcción de Laboratorios virtuales, proceso que se desarrollará a través de la generación de los

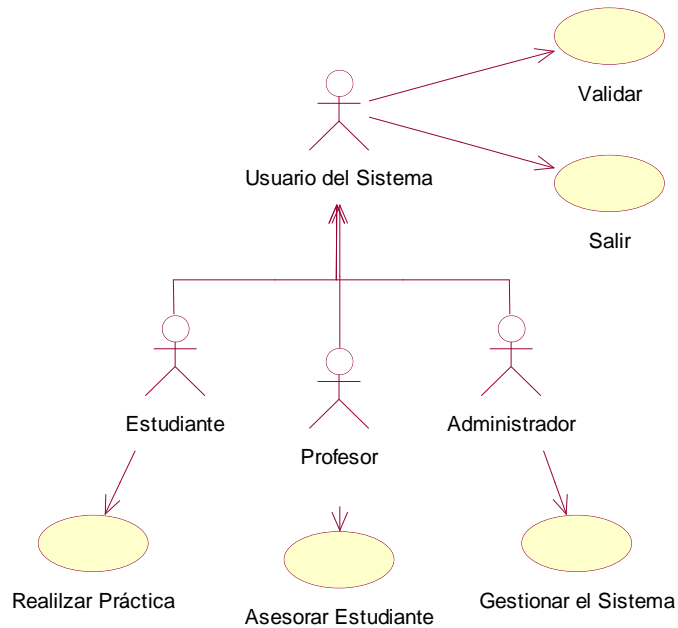


modelos exigidos por el Proceso de Modelado Unificado. En este orden de ideas se parte del modelo de casos de uso, mediante el cual se exponen las funcionalidades básicas del sistema, mientras que el modelo de análisis brinda las clases que participan en cada uno de los casos de uso. Finalmente la arquitectura de referencia establece las relaciones y responsabilidades de todos los paquetes que participan en el modelo de análisis.

## **2.2 MODELO DE CASOS DE USO**

La identificación de estos casos de uso es dirigida por la documentación revisada en los laboratorios y las funcionalidades ofrecidas por el LMS dotLRN, elementos presentados en el capítulo uno, los cuales permiten establecer los requisitos esenciales que un laboratorio virtual debe tener en una plataforma e-learning [AF01]. Debe aclararse que algunas de estas funcionalidades las soporta el sistema de gestión de e-learning y que por tanto son un complemento para que los laboratorios virtuales sean una unidad e-learning más eficiente.

En un nivel de abstracción más elevado aparecen los siguientes casos de uso genéricos, ver figura 2.1, ligados a los tres actores que se identifican en este nivel.



**Figura 2.1. Diagrama de Casos de Uso Genérico.**

Un Sistema de Gestión de Aprendizaje básicamente considera la interacción de tres usuarios: estudiante, profesor y administrativo. El estudiante es el actor que pretende adquirir las habilidades y entrenamiento mediante la ejecución de prácticas virtuales. El profesor es el actor responsable de la asesoría de las prácticas realizadas por el estudiante y seguimiento del progreso de las mismas. El administrativo es el actor responsable del buen funcionamiento del sistema, en esta caso el LMS.

Los tres actores al interactuar con el LMS necesitan de los servicios de autenticación, necesarios para asegurar que el entorno “classroom”, así como los diversos servicios ofrecidos por el LMS sean utilizados de la manera correcta.

En un nivel de abstracción inferior aparecen tres casos de uso principales, ligados a los tres actores que identificamos en el nivel correspondiente a los casos de uso genérico.

El estudiante es el actor que pretende adquirir las habilidades y entrenamiento mediante

la ejecución de prácticas virtuales (Realizar Práctica). El estudiante espera tener un entorno de aprendizaje flexible que sea fácil para navegar y facilite el aprendizaje de contenidos. Esto normalmente requiere algún tipo de personalización de la interfaz y el contenido, así los estudiantes tendrán diferentes percepciones de las herramientas que usan.

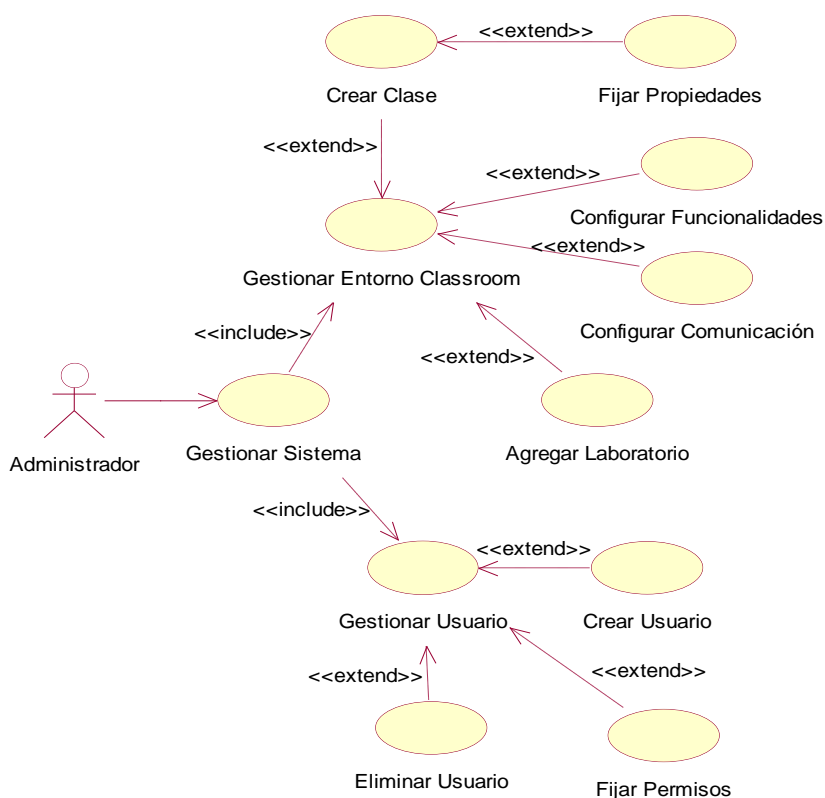
El profesor es el actor responsable de la asesoría de las prácticas realizadas por el estudiante y seguimiento del progreso de las mismas (Asesorar Estudiante). El profesor está buscando herramientas que ayuden a producir y distribuir cursos a un menor esfuerzo. Los instructores también requieren de una herramienta que ayude a medir el progreso, a través de una evaluación on-line que facilite el entendimiento del progreso de la clase y de las dinámicas de la clase.

El administrativo es el actor responsable del buen funcionamiento del sistema, en este caso el LMS (Gestionar Sistema). El administrador necesita medios para gestionar los anteriores usuarios.

Un proceso de refinamiento sucesivo de los tres principales casos de uso ligados a los tres actores que se identificaron, permitirá identificar nuevos casos de uso, los cuales corresponden a las interacciones específicas de los usuarios de un laboratorio virtual integrado a un LMS. Para empezar la figura 2.2 muestra el refinamiento del caso de uso "Gestionar Sistema". En el diagrama se observa como aparecen dos casos de uso que extienden el especificado en el caso de uso base.

Específicamente se muestra que el administrador puede crear una clase, fijando las propiedades apropiadas para ésta, como nombre, periodo de duración, y el profesor asignado. Una vez realizado lo anterior el administrador procede a configurar algunas funcionalidades propias de un entorno classroom, así como las herramientas de comunicación necesarias. Finalmente el administrador agrega el laboratorio donde se realizarán las prácticas. Todo este comportamiento se recoge en el caso de uso

“Gestionar Entorno Classroom”. Un entorno “classroom” es el término más frecuente usado en entornos e-learning, y describe diversas fuentes como chats, conferencias lecturas publicadas, materiales de cursos, etc; y ofrece servicios de un salón de clase convencional.

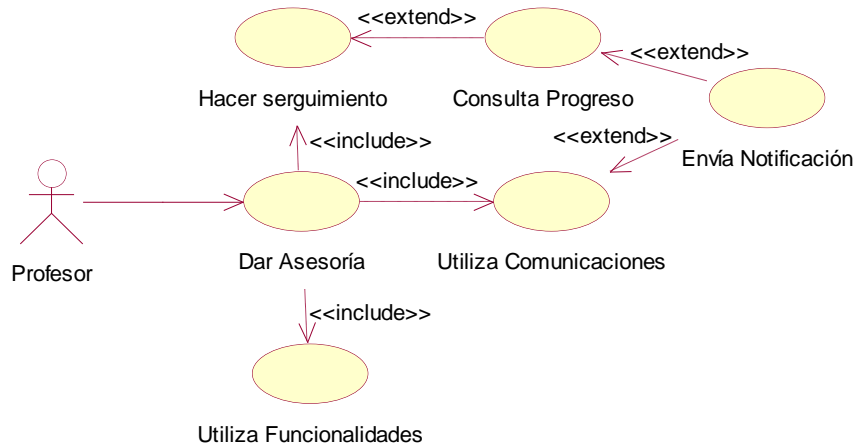


**Figura 2.2. Diagrama Extendido del Caso de Uso Gestionar Sistema.**

Por otra parte el diagrama también muestra que es el administrador, quien se encarga de crear un nuevo usuario, asignarle sus respectivos permisos y finalmente eliminarlo del sistema; este comportamiento se refleja en el caso de uso “Gestionar Usuario”.

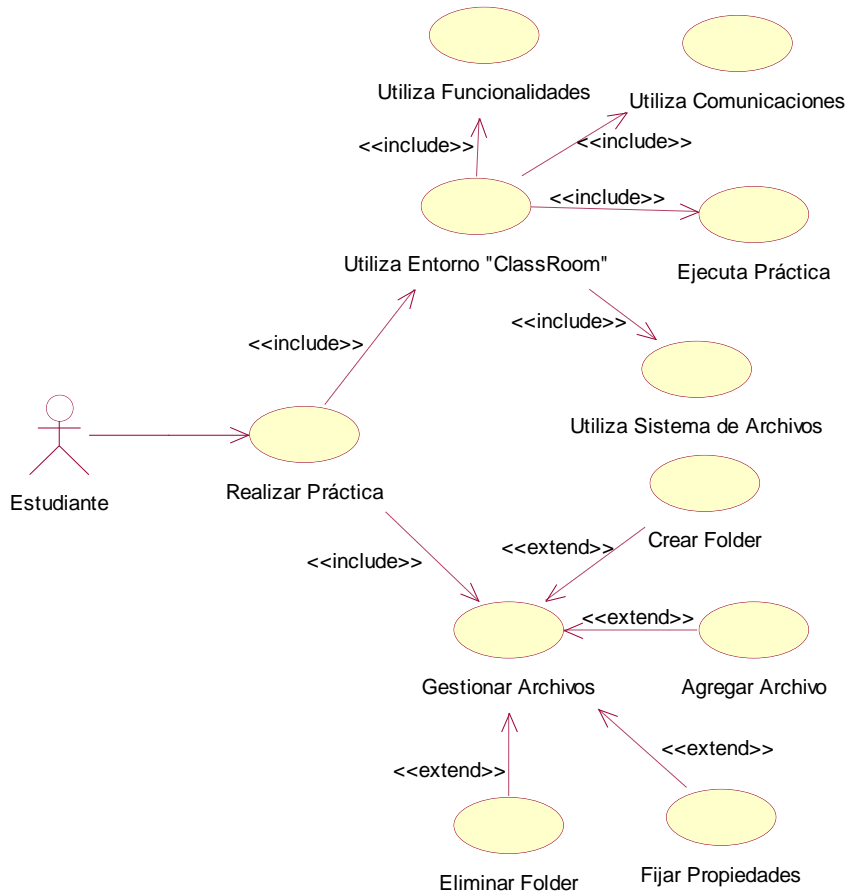
Siguiendo el proceso de refinamiento sucesivo, la figura 2.3 muestra el refinamiento del caso de uso “Asesorar Estudiante”. El profesor es el encargado de realizar las asesorías respectivas en las prácticas realizadas por los estudiantes, para lo cual consulta el progreso y avance de éstos en las diversas sesiones de trabajo, para

posteriormente a través de las herramientas de comunicación y demás funcionalidades del entorno “classroom”, procede a enviar notificaciones que aclaren dudas e inquietudes.



**Figura 2.3 Diagrama Extendido del Caso de Uso “Asesorar Estudiante”**

La figura 2.4 muestra el refinamiento del caso de uso “Realizar Práctica”, uno de los más importantes. A través de este caso de uso, un estudiante puede hacer uso de todos los servicios ofrecidos por el entorno “classroom” para la ejecución de prácticas de laboratorios virtuales.



**Figura 2.4. Diagrama Extendido del Caso de Uso “Realizar Práctica”**

El estudiante no solo puede ejecutar las prácticas de laboratorio, si no que también puede interactuar con otros estudiantes haciendo uso de las herramientas de comunicación, así como hacer uso de las demás funcionalidades que el administrador configuró previamente.

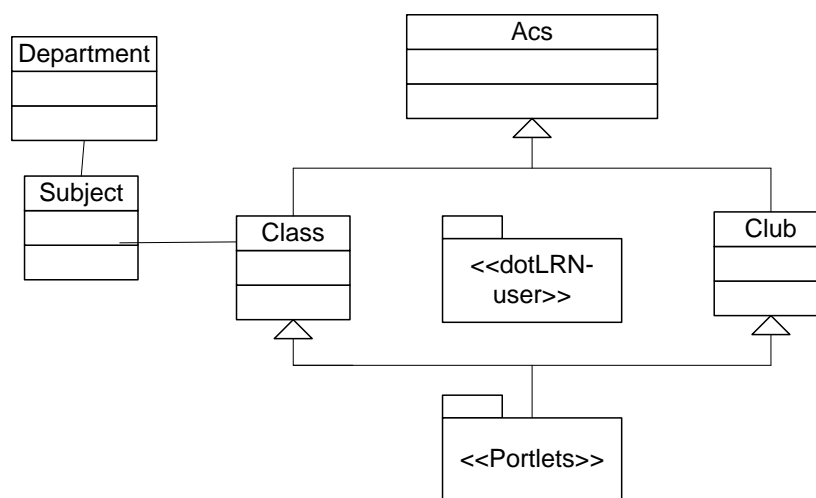
### 2.3 MODELO DE ANÁLISIS

Los casos de uso son realizados mediante colaboraciones de clases. En primer lugar es necesario identificar aquellas clases que formarán el modelo de análisis. Estas clases son identificadas utilizando los casos de uso definidos previamente. A partir de

un número reducido de estos casos de uso se identificarán las clases que puedan implementarlos.

La figura 2.5 representa el diagrama de clases para la realización del caso de uso “Gestionar Sistema”. Este caso de uso esta representado básicamente por los casos de uso gestionar entorno classroom y el caso de uso gestionar usuario.

De manera específica, el diagrama representa las clases que participan en la creación y configuración de un entorno “classroom”, correspondiente al caso de uso “Gestión de Entorno Classroom”



**Figura 2.5. Diagrama acs-subsite**

Explícitamente la figura 2.5 muestra en un diagrama de clases cómo las comunidades de dotLRN pueden ser clases o clubes, ambos tipos proporcionan juegos diferentes de herramientas de colaboración, los roles para personas y plantillas con valor predeterminado. En ambos casos, estas comunidades tienen su propia instanciación de calendarios, foros, las noticias, los grupos, el archivo compartido, etc. La figura 2.5 también muestra esta estructura y cómo las clases relacionan el departamento con la

clase subject. “los departamentos” describen la estructura orgánica normal dentro de la institución, “el Subject” contiene información que repite en los diferentes casos de un “Class.” “Class” especifica personas y sus papeles, salida y la fecha de acabado, así como otros atributos.

En el e-learning las comunidades sirven para compartir un espacio de trabajo, en el cual varios integrantes de la comunidad pueden aprender a través de los cursos que toman de la comunidad y aportar a ésta sus estudios. Tales comunidades pueden representar grupos de interés especiales, proyectos o simplemente organizaciones sociales. Implementar este tipo de actividades en dotLRN requiere definir un alcance muy claro de cada aplicación, y así cada course/community puede tener sus propias aplicaciones fijas, plantillas y permisos. Para lograr esto dotLRN usa el concepto “subsites” que separa los contextos entre los diferentes tipos de comunidades (“las clases” y los “clubes”, para nosotros asignaturas y comunidades). Cada caso del subsite (“dotLRN-community”) puede tener su propia estructura de permisos.

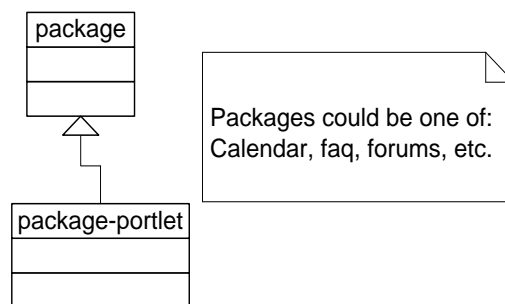
El paquete workflow provee el soporte necesario para la implementación de un servicio de seguimiento de un proceso, alrededor de un objeto. Este servicio mantiene el rastro de un proceso que se desea seguir, el estado normal de ese proceso y quien ejecuta ese estado. En general, un workflow consiste de un set de estados, y un set finito de acciones. El paquete está diseñado para soportar workflows basados en un modelo correspondiente a una maquina de estados finitos. El objetivo de integrar los laboratorios virtuales a este servicio es aprovechar las capacidades de seguimiento que ofrece y así poder generar una retroalimentación que suministre información sobre la situación del estudiante en el desarrollo de las prácticas.

Los workflow habilitan a las aplicaciones para usar todas las alertas y las capacidades de gestión de eventos construidas en el paquete del mismo nombre. El sistema workflow podría colocar los correos electrónicos para ser enviados cuando una tarea ha sido terminada o se necesita empezar, podría poner en orden las tareas para mostrar



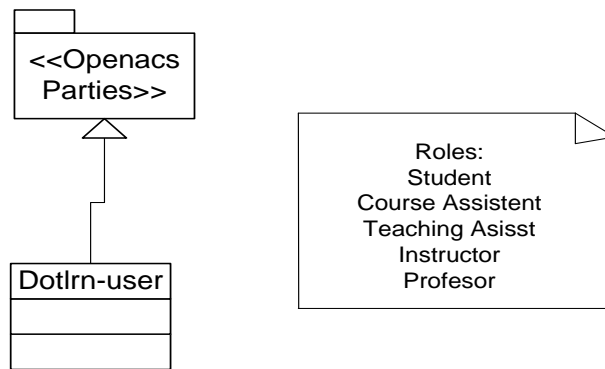
en los usuarios, centralizar el calendario o podría proporcionar rasgos que no fueran considerados en el momento de diseñarlos en las aplicaciones.

La figura 2.6 representa el diagrama de clases para la realización del caso de uso “Realizar Práctica”. Este diagrama representa en concreto el caso de uso “Agregar Práctica”, y que constituye la integración de un laboratorio a dotLRN. dotLRN utiliza la arquitectura del portal OpenACS para generar la interfaz del usuario para las diversas aplicaciones que se integran al LMS, lo cual incluye a los laboratorios virtuales.



**Figura 2.6 Diagrama portlet**

Los portlets de dotLRN generan la interfaz apropiada entre las aplicaciones de OpenACS (para el caso, los laboratorios virtuales) y los portales de dotLRN; haciendo que, estos portlets entreguen funcionalidad personalizada basada en las aplicaciones de OpenACS; de esta manera se hace la integración en dotLRN no solo de los laboratorios virtuales, sino de todas las herramientas que se necesitan para un entorno classroom. Este diagrama representa las clases que participan en el caso de uso



**Figura 2.7 Diagrama dotLRN-user**

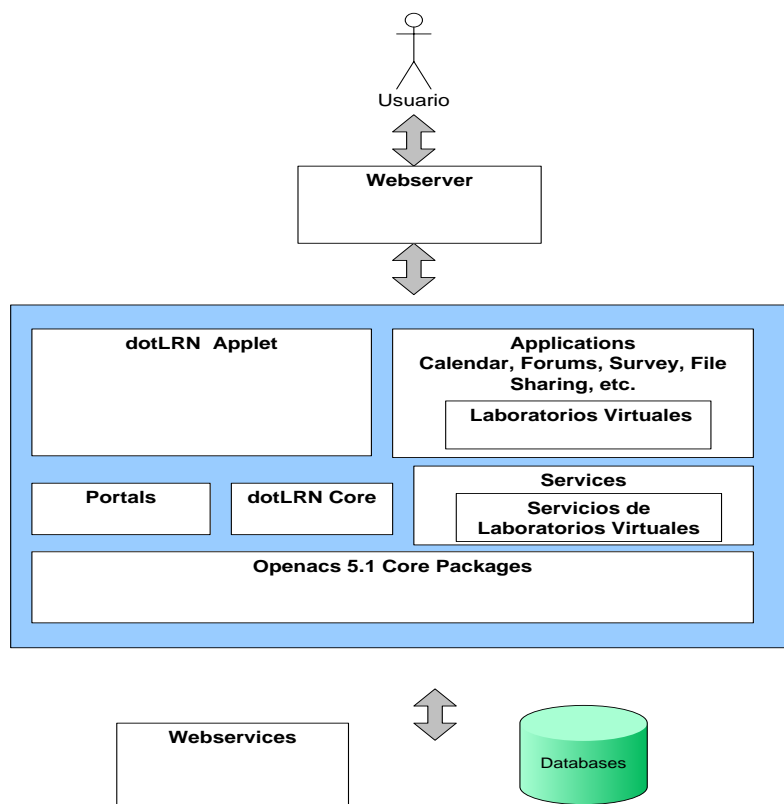
OpenACS puede soportar complejas jerarquías de usuario, definiendo parties, esto puede ser: “los usuarios” identificados por un distinto correo electrónico, o una “persona” con el distintivo primero y el último nombre, o un “grupo” ó incluso otros grupos. Con esta arquitectura de dotLRN, los usuarios no solo pueden tener más de un rol en particular, sino una jerarquía de permisos. El diagrama de la figura 2.7 representa las clases que participan en el caso de uso “Gestión de Usuario”.

El paquete XML-RPC merece ser mencionado por cuanto es el componente que permite interoperar con otras plataformas y sistemas operativos. XML-RPC es un protocolo que permite a los clientes hacer llamadas a procedimientos remotos sobre su servidor. El dato es transformado dentro de un formato XML antes de ser transferido entre el cliente y el servidor. Esto permite al software interactuar usando diferentes sistemas operativos y lenguajes de programación. Está diseñado para ser tan simple como sea posible, transmitir, procesar y retornar complejas estructuras de datos.

## **2.4 ARQUITECTURA DE REFERENCIA**

El establecimiento de una arquitectura de referencia se ha realizado identificando subsistemas como agrupaciones de clases del modelo de análisis. El criterio que se ha

seguido es el de agrupar elementos que tiendan a cambiar juntos, tanto como por los modelos de información que manejan como por mejoras de versiones. Así mismo, no todos los subsistemas de un elemento del modelo de referencia son obligatorios. La adición de subsistemas puede mejorar los niveles de servicios ofrecidos.



**Figura 2.8. Arquitectura de referencia para la implementación de laboratorios virtuales en dotLRN**

La figura 2.8 muestra la arquitectura de referencia propuesta para la implementación de laboratorios virtuales en dotLRN con la cual se busca la integración y reutilización de los diversos servicios de e-learning que la plataforma dotLRN ofrece, así como la implementación de algunos que la misma plataforma no contempla y que sean necesarios con el ánimo de generar un mayor valor agregado al LMS a la hora de

entregar un laboratorio virtual como una aplicación e-learning.

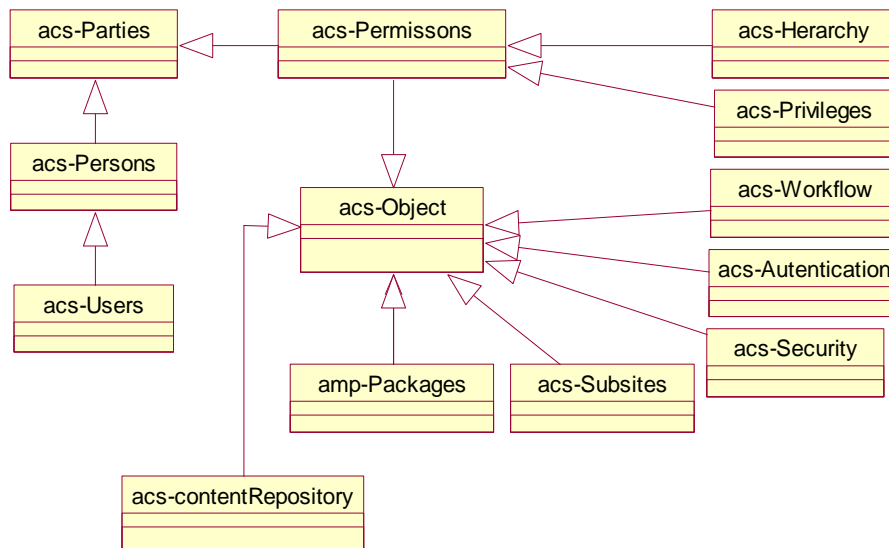
La funcionalidad de la arquitectura propuesta es ofrecida por el concurso de diversos componentes. Cada componente está dirigido a proveer una característica encontrada típicamente en las aplicaciones interactivas y colaborativas soportadas por la plataforma.

Una vez fijadas las características generales de un LMS se hace necesario centrar la atención de manera específica en dotLRN como el LMS del framework openACS, específicamente en sus funcionalidades generales que le darán valor agregado a un Laboratorio Virtual como aplicación de aprendizaje, así como también en aquellos paquetes que permitirán la integración de estas.

La arquitectura que se muestra en la figura 2.8 enseña una arquitectura funcional basada en múltiples capas que se integran a la arquitectura de OpenACS y dotLRN, las cuales se describen a continuación:

#### **2.4.1 OpenAcs-Core**

En esta capa se encuentra la funcionalidad básica de OpenACS. Aquí se encuentran los paquetes del núcleo, ver figura 2.9, que se requieren para que OpenACS funcione. Estos incluyen un Kernel, gestión de paquetes, utilidades para la administración de los usuarios, el sitemap, otros objetos comunes, un sistema de plantillas que separa claramente la lógica de negocio de la presentación, y un sistema de mensajería y correo backend [Open].



**Figura 2.9. Core OpenACS**

Los servicios como el paquete de eventos son aplicaciones de lógica reusable y no son ninguna aplicación dependiente ni requieren una interfaz del usuario. Los paquetes que se denominan aplicación tienen una interfaz de usuario y su código está claramente dividido en la lógica de aplicación y la lógica de presentación.

Cada servicio y aplicación requieren un modelo de datos que se integrará en el framework. La integración de estos modelos de datos permite la reusabilidad y diferentes funcionalidades. Para aumentar la reusabilidad, el framework tiene una arquitectura orientada a objetos, los cuales deben ser añadidos por el framework.

Los objetos de OpenACS tienen sus atributos guardados en tablas y un juego de métodos definidos como los paquetes de PL/SQL. Estos paquetes contienen los procedimientos que proporcionan la interfaz para el modelo de datos.

La idea que hay detrás de la arquitectura orientada a objetos de OpenACS es que cada pedazo de información que puede ser reusable debe ser un objeto. Las estructuras de datos se integran en el framework creando una tabla llamada "acs-objects" que guarda

un registro de cada objeto de OpenACS. Otra tabla llamada “acs-object”, define los atributos generales guardados en cada objeto, incluso un único ID, un tipo del objeto. El concepto de los tipos de objetos en OpenACS es equivalente a las clases en el paradigma orientado a objetos de los lenguajes de programación, con la excepción de que al utilizarse un RDBMS, debe realizarse un trabajo adicional.

Aunque la reusabilidad es la meta principal, gracias a este diseño, el sistema de permisos puede ayudar al diseñador a definir que se permite hacer a los usuarios, esto está definido en las filas de la tabla para cada aplicación. Cada objeto tiene un atributo llamado “context-id” que especifica los permisos predefinidos para un objeto y su alcance.

Se definen los tipos del objeto con atributos definidos en el `acs_object_types` y `acs_attributes`. Los tipos del objeto en OpenACS pueden heredar los atributos de un tipo del padre, por lo tanto los tipos del sistema son jerárquicos. En conclusión el modelo de datos en OpenACS se lleva a cabo en un RDBMS y esta herencia necesita ser realizada explícitamente.

El sistema de permisos de OpenACS permite a diseñadores y a administradores poner políticas de control de accesos a nivel del objeto, es decir, cualquier aplicación u objeto del sistema representado por una fila en la tabla `acs_object` puede controlarse vía un simple PL/SQL o interfaz de Tcl. El sistema de permisos maneja un modelo de datos que permite mediante scripts verificar permisos usando una simple llamada al API. Adicionalmente, openACS, tiene dos mecanismos auxiliares, el sistema de Grupos, el cual permite que los usuarios se agrupen de manera flexible y la noción de “contexto del objeto” que permite agrupar objetos en dominios de seguridad más grandes. En resumen el modelo de datos de permisos es bastante simple, y se constituye en un mapeo entre privilegios, parties y objetos. Los privilegios son unidades básicas para implementar las políticas de control de acceso. En OpenACS, la tabla de privilegios está organizada jerárquicamente para que los diseñadores puedan definir conjuntos de

privilegios y agregarlos juntos.

En openACS cada pedazo es registrado y almacenado en un “Content Repository” que estandariza la manera en que las aplicaciones acceden y maneja esta información. Esta capa adicional de complejidad hace posible la reusabilidad de todas las funcionalidades de gestión de contenidos como los permisos, formatos u otras funcionalidades diferentes que no pueden haber estado incluidas en el momento de diseñar una aplicación. Un servicio creado para el Content Repository, como una recuperación de información o el sistema de clasificación de documentos automático, puede usarse automáticamente por todas las aplicaciones en el sistema.

Finalmente el servicio workflow de OpenACS agrega capacidades útiles a las aplicaciones, habilitándolas para usar todas las alertas y las capacidades de gestión de eventos construidas en el paquete workflow.

#### **2.4.2 Capa de Servicios**

En esta capa se encuentran los servicios comunes a todas las aplicaciones que se puedan contemplar como “laboratorios virtuales”. Estos servicios son el resultado de la abstracción de los requisitos básicos de un laboratorio virtual. La capa de servicios de la arquitectura es responsable de proveer los componentes software con la funcionalidad colaborativa subyacente [Open].

Los servicios son paquetes que no interactúan con el usuario y establecen una interfaz entre los paquetes, donde reside la aplicación real. Como dotLRN se constituye en el LMS de OpenACS, entonces hereda todas las funcionalidades de OpenACS, incluyendo modularidad y demás características.

En la sección 2.4.4 se describen los principales servicios que un LMS podría ofrecer

para un entorno de laboratorios virtuales.

### **2.4.3 Capa de Aplicaciones**

Finalmente, el nivel de aplicación, se encuentran las aplicaciones consideradas como laboratorios virtuales, que serán entregadas al usuario; los usuarios acceden a estas mediante interfaces Web interactivas. Esta capa se caracteriza por hacer uso de los diversos servicios ofrecidos por la capa inmediatamente inferior.

DotLRN ofrece varias aplicaciones colaborativas como son los calendarios, los foros, los ficheros compartidos, herramientas de valoración, etc. Eso sumado a los que el almacén de openACS proporciona (es decir, gestión de contenidos, workflow, webmail, etc). Muchas de estas funcionalidades son similares a las encontradas en otros LMS como WebCT y Pizarra. En otras palabras dotLRN usa el framework orientado a objetos de OpenACS para la creación de aplicaciones web, soportado en el modelo de datos estándar y en el API de OpenACS.

### **2.4.4 Servicios de Laboratorios Virtuales**

A continuación se describen las funcionalidades de los servicios ofrecidos por el módulo de laboratorios virtuales dentro de la plataforma dotLRN. Cada uno de los servicios contempla múltiples casos de uso para el éxito del mismo. Los servicios que se muestran a continuación se consideran básicos.

#### **Servicio de Entorno “Classroom”**

Con este servicio se busca dar la posibilidad de gestión y conformación de entornos virtuales para la realización de prácticas en el marco de los laboratorios virtuales seleccionados. Este servicio debe brindar la posibilidad de dar acceso solamente a los usuarios registrados, a través de un login y un password e indicar si el aula está abierta o cerrada. Los tutores serán los moderadores de las respectivas aulas.



## **Servicio de Seguimiento**

En la distancia, se hace necesario, que los estudiantes no se sientan solos y los profesores conocer lo que los estudiantes están haciendo. En la educación tradicional, para los profesores es muy fácil conocer si los estudiantes están aprendiendo en la forma correcta. Esto no es fácil en los entornos de e-learning. Para suplir esta funcionalidad, un aplicación de laboratorio virtual en dotLRN debe ayudarse de un sistema de monitoreo que le permita seguir la interacción de los estudiantes con la aplicación (laboratorio).

La mayoría de los profesores experimentados obtienen información acerca de las acciones alcanzadas sobre el laboratorio y tienden a ser consideradas como importantes desde un punto de vista pedagógico. Estas acciones serán reflejadas en el seguimiento del estudiante. Si un estudiante alcanza una tarea dada , un reporte es enviado al tutor responsable de monitorear sus acciones. Luego sin estar en el mismo salón de clase, el profesor es capaz de seguir el desempeño del estudiante y si es necesario, enseñar como hacer la práctica de entrenamiento utilizando las herramientas de comunicación.

El servicio de monitoreo se puede catalogar así:

### **a) Servicio de Seguimiento Síncrono**

Este servicio pretende implantar un sistema de seguimiento síncrono que permita al tutor del aula virtual un conocimiento de las acciones realizadas por los estudiantes en el desarrollo de la práctica y así poder generar un proceso continuo de retroalimentación.

### **b) Servicio de Seguimiento Asíncrono**

Dado que en determinadas circunstancias no es posible realizar un seguimiento síncrono por parte del tutor, se hace necesario que la plataforma ofrezca el servicio de seguimiento asíncrono, mediante el cual las acciones realizadas por el estudiante durante la práctica queden almacenadas y posteriormente pueden ser analizadas por el tutor para una respectiva retroalimentación.

### **Servicio de Comunicaciones**

Los sistemas de aprendizaje basados en Internet necesitan suplir a los estudiantes y profesores con herramientas que les permitan interactuar con otros. El modelo de laboratorios virtuales está diseñado para dar a los estudiantes e instructores las herramientas de comunicación necesarias y así conseguir un alto grado de competencias y destrezas cognitivas. Para ello se incluyen diversas herramientas tales como, Chat, foros, correo electrónico, etc.

Estas herramientas pueden ser usadas por los profesores para enviar instrucciones de la práctica a sus estudiantes, y por los estudiantes para retornar sus resultados y conclusiones, para compartir experiencias a otros estudiantes y enviar comentarios al gestor del sistema.

### **Servicio de Archivos**

Este servicio busca brindar la posibilidad de guardar las prácticas realizadas por los usuarios durante las sesiones de uso del laboratorio virtual.

### **Servicio de Gestión de Usuario**

Este servicio busca dar gestión a las capacidades de los usuarios del laboratorio virtual, tales como permisos de acceso, roles, prácticas realizadas, etc.

### **III. DISEÑO DE UN PROTOTIPO DE LABORATORIO VIRTUAL.**

El objetivo de este capítulo es describir el proceso de diseño e implementación de un prototipo de laboratorio virtual que permita la validación de la “Aplicación del paradigma learning by doing mediante la construcción de laboratorios virtuales en la plataforma dotLRN del proyecto E-LANE en Universidad del Cauca”; arquitectura propuesta y descrita en el capítulo dos. Para tal fin, se seleccionó el diseño e implementación del prototipo Laboratorio Virtual AXE-10 (LVAXE-10) para el curso Laboratorio II de Sistemas Telemáticos, que se desarrolla en la Central AXE-10 del Departamento de Telemática de la FIET, el cual según la bibliografía revisada se constituye en un primer prototipo que se integra al LMS dotLRN.

En este orden de ideas se procedió a categorizar tal laboratorio, recordando el capítulo uno, en el cual se estableció que el paradigma learning by doing comprende dos grandes aproximaciones; los sistemas de aprendizaje práctico basados en acceso

remoto a sistemas reales, donde estos pueden ser laboratorios hardware y laboratorios software; y por otra parte los sistemas de aprendizaje prácticos basados en laboratorios virtuales. Esta clasificación permitió ubicar la central AXE-10 en la primera categoría por cuanto se entiende como un sistema real al cual se desea acceder remotamente a través del LMS.

Hecha esta ubicación, el capítulo presenta a consideración del lector cuatro de los cinco modelos que describen el proceso de desarrollo del Laboratorio Virtual AXE-10, el Modelo del Ambiente, el Modelo de Análisis, el Modelo de Diseño y el Modelo de Despliegue.

El propósito del Modelo Inicial del Ambiente es encontrar las entidades del negocio relevantes y los casos de uso del sistema que deberán ser soportadas por él, modelando justo lo necesario para entender el contexto en el cual será implantado el sistema.

El Modelo de Análisis del Laboratorio Virtual AXE-10 busca identificar el conjunto de clases que permite la realización del modelo de casos de uso completo por medio de colaboraciones. Por otra parte el Modelo de Diseño del proceso se elaboró utilizando el Modelo de Análisis como punto de partida, el cual es adaptado al entorno que se ha seleccionado para la implementación.

Por otra parte el Modelo de Despliegue se elaboró a partir del establecimiento y descripción de los nodos y configuraciones en red, elementos que se dedujeron de los diferentes subsistemas establecidos en el anterior modelo.

Finalmente los anexos presentan en forma detallada la metodología que se siguió para tal propósito, mientras que el presente capítulo corresponde a la descripción esencial de la misma.

### 3.1 DESCRIPCIÓN FUNCIONAL DEL LABORATORIO VIRTUAL AXE-10

El Laboratorio Virtual AXE-10 para la materia Sistemas Telemáticos II de la Tecnología en Telemática de la FIET, desarrollado en la Central AXE-10, tiene el objetivo de validar la arquitectura propuesta descrita en el capítulo anterior. Su utilidad se enmarca en la realización de prácticas virtuales dentro del curso Laboratorio II de Sistemas Telemáticos de la tecnología mencionada, que se desarrolla en la Central AXE-10, sistema real al cual se accede remotamente desde el LMS dotLRN.

El objetivo principal para desarrollar el LVAXE-10 es aportar al desarrollo del proceso de descentralización que esta llevando la Universidad, además de permitir a los estudiantes de los diferentes municipios donde se esta llegando, acceder de forma remota a la Central digital de Enseñanza Ericsson AXE-10; con esto se evita que los estudiantes de estas regiones apartadas se desplacen hasta el lugar donde esta la central para desarrollar sus practicas. Por otro lado los estudiantes al interactuar remotamente con la central, logran tener una idea previa de lo que encontrarán en las prácticas que se realizan localmente. Esto se traduce en un ahorro significativo de recursos y tiempo en el proceso de aprendizaje.

En el desarrollo *normal* del laboratorio II de Sistemas Telemáticos, que se dicta en la Tecnología en Telemática, se pretende entre otras cosas, que el estudiante conozca la estructura física de la central, la organización de sus diferentes módulos, la funcionalidad de sus dispositivos con las diferentes operaciones que estos realizan; así como la gestión, operación y mantenimiento que se puede realizar a la Central a través de comandos por medio de un terminal, aspecto de gran importancia.

Un análisis de las temáticas impartidas en el curso del Laboratorio II de Sistemas Telemáticos muestra que se compone de una parte expositiva, cuyas actividades se pueden impartir por medio de clases magistrales y una parte práctica que por el contrario necesita exclusivamente que el estudiante interactúe con el hardware de la Central Ericsson para el logro de los objetivos de la asignatura; es en esta parte del

proceso de aprendizaje donde el laboratorio virtual aporta de manera significativa.

En este orden de ideas se desarrollo el prototipo LVAXE-10, en el cual se introdujeron conceptos básicos del desarrollo de esta práctica. Con el LVAXE-10 se pretende que el estudiante conozca la constitución física de la central, como también que pueda ejecutar remotamente comandos para la gestión de la misma. Para lograr estos objetivos la aplicación desarrollada puede ser descrita en dos partes:

a) Visualización del hardware de la Central

El objetivo de este componente es lograr un acercamiento del estudiante con la construcción física de la central, para esto se elaboró una presentación flash que recoge los principales componentes de la estructura física de la Central Ericsson AXE-10 con su respectiva descripción, esto contribuye a que el estudiante se familiarice con el hardware, con el cual posteriormente va a interactuar; además de brindar un componente didáctico que asegure el proceso de aprendizaje.

b) Gestión de comandos.

En la materia Laboratorio de Sistemas Telemáticos II la parte correspondiente a prácticas en el manejo de comandos se hace a través de terminales contiguos a la Central, los cuales están conectados a través del puerto serial y utilizan la interfaz gráfica Winfiol, que permite enviar comandos y recibir la respectiva respuesta de la Central. Una vez establecida la conexión con la Central, el estudiante puede ejecutar comandos para habilitar y deshabilitar abonados, localizar hardware, así como también gestionar las categorías de los diferentes servicios especiales que la Central ofrece a los abonados. Una de las prácticas que se realizan en el curso, corresponde a la gestión de servicios especiales de abonados; la asignación y cancelación de tales servicios se realiza a través de comandos específicos para cada servicio, según lo especifique la guía del laboratorio. Esta práctica se puede comprobar utilizando los teléfonos que se encuentran conectados directamente a la Central.

Partiendo de esta situación, el LVAXE-10 busca recrear el mismo entorno de aprendizaje real, a través de la implementación de diferentes interfaces que emulan los diversos pasos de la práctica. El primer aspecto del LVAXE-10 es darle a conocer al estudiante que se va a conectar a la central remotamente, enfatizando de esta forma la conexión remota. En la práctica real la selección de los abonados con los cuales se trabaja se hace de mutuo acuerdo en coordinación con el docente, pero en el LVAXE-10, una vez el estudiante se conecta a la central es conveniente permitir la selección de los abonados y deshabilitarlos para que otro usuario no los seleccione evitando posibles conflictos. Después de seleccionar los abonados el estudiante puede enviar comandos y recibir respuesta de la Central, con el objetivo de que pueda gestionar las categorías de los abonados seleccionados. Se debe mencionar que esta interfaz que se le presenta al estudiante puede ejecutar cualquier comando, ya que está conectado a la Central remotamente.

Finalmente se le suministra al estudiante una interfaz de validación para que compruebe las categorías gestionadas, simulando la utilización de los abonados que se encuentran conectados a la Central.

Para asegurar el éxito de LVAXE-10 se necesita hacer una entrega didáctica de los anteriores pasos, lo cual se hace a través de la implementación de una máquina de estados finitos, la cual se encarga de guiar al estudiante durante el desarrollo de la práctica.

Para completar la validación de la arquitectura propuesta el componente práctico anterior debe acompañarse de un componente que permita realizar un seguimiento a las prácticas desarrolladas en términos de los pasos anteriormente descritos y de los servicios que el LMS puede proveer, tales como foros, chats, calendario, notas, tablero, etc.

En resumen, el laboratorio virtual consta de un componente práctico que es entregado didácticamente a través de una máquina de estados finitos, un componente de

seguimiento que le permite al docente conocer las prácticas desarrolladas por los estudiantes, y la utilización de los servicios que el LMS le provee.

Una descripción detallada de las interfaces que implementan esta funcionalidad se describen en la parte final de este documento.

## **3.2 MODELO DEL AMBIENTE**

El propósito del Modelo del Ambiente es encontrar las entidades de negocio relevantes y los casos de uso del sistema que deberán ser soportadas por él, modelando justo lo necesario para entender el contexto en el cual será implantado el sistema. Siguiendo esta dirección y con los actores definidos, así como el conjunto identificado de casos de uso, se procedió a construir un Modelo del Ambiente. Éste está constituido por los Actores, los Casos de Uso, las relaciones entre los casos de uso y la descripción inicial del escenario para cada Caso de Uso.

### **3.2.1 Descripción de Actores**

Para definir el propósito de un laboratorio se debe saber para quien está diseñado; en este caso las personas que van a interactuar directamente con el sistema son: los estudiantes, y profesores; los cuales se asimilan como los principales usuarios del sistema. De igual el laboratorio virtual AXE-10 se soporta en la Central Ericsson AXE-10, sistema considerado un actor que responde a las peticiones realizadas por los dos usuarios señalados.

Estudiante: Es el actor que pretende adquirir una formación mediante el desarrollo de prácticas virtuales. El laboratorio virtual AXE-10 está dirigido principalmente a los estudiantes que se encuentran dentro del programa de tecnología en Telemática de la FIET, de los diferentes municipios del departamento del Cauca, donde la Universidad del Cauca hace presencia en desarrollo del proceso de descentralización.



Profesor: Es el actor que colabora con el proceso de formación a través de asesorías y el seguimiento a las prácticas desarrolladas por los estudiantes. Sus acciones al interactuar involucran apoyo al estudiante mediante el conocimiento del progreso del estudiante en el proceso de formación.

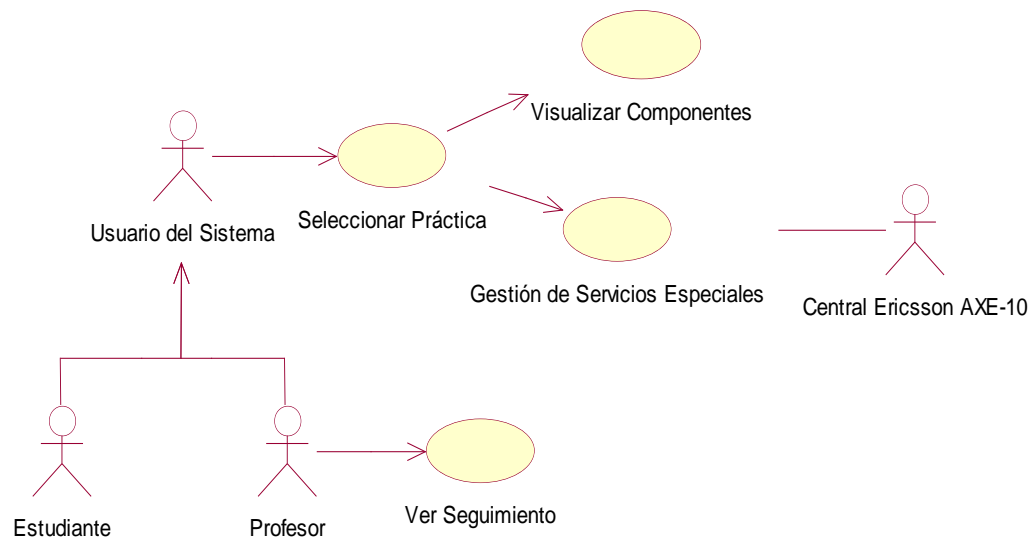
Central Ericsson AXE-10: El sistema AXE es un sistema "Stored Program Control" (SPC), lo cual significa que programas almacenados en un computador controlan la operación de la central. El sistema AXE está formado por dos partes principales: el equipo de conmutación para la conexión de la llamada telefónica y un computador para controlar el equipo de conmutación.

### **3.2.2 Modelo de Casos de Uso del Sistema**

#### **a) Diagrama de Casos de Uso del Sistema**

Teniendo presente que el objetivo central es implementar un laboratorio virtual para la materia Sistemas Telemáticos II

de la Tecnología en Telemática de la FIET, desarrollado en la Central AXE-10, y así validar la aplicación del paradigma learning by doing en el proyecto E-LANE, es necesario tener presente la visión de la funcionalidad que debe ofrecer la arquitectura de referencia del sistema. Para alcanzar el modelo de referencia de dicha arquitectura del Laboratorio Virtual AXE-10, se partió de la identificación de los casos de uso, los cuales consideran las funcionalidades esenciales. A su vez la identificación de los casos de uso permitió establecer claramente tres componentes del laboratorio, los cuales se corresponden con tres escenarios necesarios para la ejecución de prácticas. Tales escenarios se muestran en la figura 3.1, en función de sus respectivos casos de uso.



**Figura 3.1 Servicio de Práctica - Gestión de Servicios Especiales, Visualización de Componentes y Seguimiento de Prácticas**

El desarrollo del Laboratorio Virtual AXE-10, comprende la interacción de tres componentes, cuya funcionalidad conjunta dan soporte al proceso de aprendizaje práctico necesario en un entorno e-learning. El componente visual permite al estudiante un conocimiento virtual tanto general, como específico del sistema al cual está accediendo remotamente; mientras que el componente práctico le brinda al estudiante la posibilidad de interactuar directamente con el sistema, en este caso la Central AXE-10, a través del envío de comandos. Finalmente el componente de seguimiento de prácticas de forma complementaria permite al profesor conocer las diversas interacciones que el estudiante realizó en el desarrollo de las mismas. Estos componentes se corresponden con tres escenarios que se describen de manera resumida a continuación:

**b) Descripción de los Escenarios de los casos de uso.**

**Escenario 1.** Descripción de los casos de uso para el Servicio Ejecutar Práctica –

## Gestión de Servicios Especiales – Componente Práctico.

Para un adecuado proceso de aprendizaje práctico, el Laboratorio Virtual AXE-10 presenta un componente práctico que ofrece la facilidad de interactuar en forma directa con el sistema real al cual se está accediendo. En este escenario el estudiante, a través de la interfaz debe seleccionar la opción correspondiente a la gestión de servicios especiales; comportamiento que se implementa en el caso de uso “Seleccionar Práctica”. Una vez el estudiante ha tomado esta decisión, el sistema se encarga de hacer la conexión al sistema real, en este caso la Central AXE-10, y le muestra la interfaz principal la cual lo invita a configurar la práctica. Para lograr esta configuración el estudiante procede a fijar los abonados con los cuales trabajará en el desarrollo de las prácticas. Después de haber seleccionado los abonados (A, B, C) con los cuales hará la práctica, procederá a escribir y enviar correctamente la sintaxis del comando, hacia la Central AXE-10, la cual hace el respectivo proceso que será retornado en una caja de texto de la misma interfaz.

Finalmente para validar el estudiante procede a realizar una comprobación de las categorías gestionadas por los comandos que envió, a través de la respectiva interfaz que le presenta tres abonados.

## **Escenario 2** Descripción de los casos de uso para el Servicio Ejecutar Práctica – Visualización de Componentes- Componente Visual.

El Laboratorio virtual propuesto necesita de un componente visual, los cuales se implementan con los casos de uso “Seleccionar Práctica”, y “Ver Componente”. El Caso de uso “Seleccionar Práctica” le permite al estudiante acceder a la Visualización de Componentes. Por otra parte el estudiante o el profesor después de seleccionar la opción Visualización de Componentes, procede a fijar uno de los elementos que conforman físicamente la Central AXE-10, con el fin de observar la posición física de los

mismos, y así tener un conocimiento del sistema al cual está accediendo, este comportamiento se implementa en el caso de uso “Ver Componente”.

Finalmente para el estudiante no sólo es posible ver la posición de ese componente, sino que puede tener acceso a una descripción conceptual del mismo, lo cual colabora con el proceso de aprendizaje.

**Escenario 3** Descripción de los casos de uso para el Servicio Ejecutar Práctica – Seguimiento de Prácticas- Componente Didáctico.

Este componente está habilitado sólo para los profesores, y su objetivo es permitir al docente conocer el desarrollo de las prácticas en términos de sus posibles estados y las acciones ejecutadas en cada una de ellos. Este se implementa con los casos de uso “Seleccionar Práctica” y “Ver seguimiento de prácticas”.

### **3.3 MODELO DE ANÁLISIS DEL SISTEMA**

El Modelo de Análisis del MCS identifica el conjunto de clases que permite la realización del modelo de casos de uso completo por medio de colaboraciones. Para el Laboratorio Virtual AXE-10 se identificaron tres paquetes básicos siguiendo el patrón de diseño Vista – Control - Modelo, los cuales soportan la funcionalidad del mismo. A continuación se describen los paquetes y las clases que dichos paquetes contienen, así como los diagramas de las mismas que soportan la funcionalidad de la aplicación.

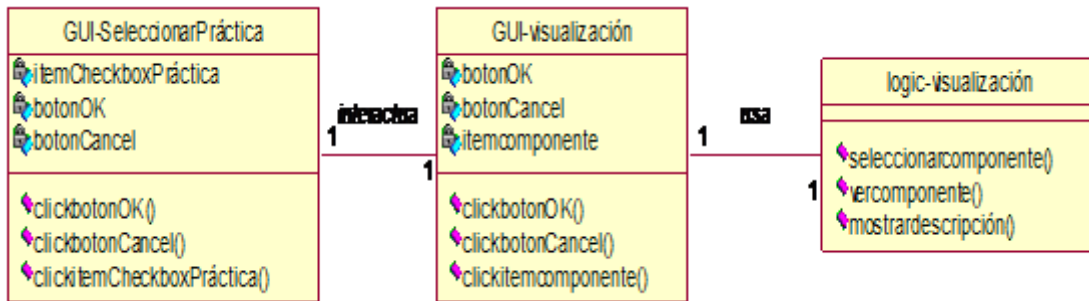
#### **3.3.1 Diagramas de Clases del Servicio**

A partir de los casos de uso obtenidos se elaboraron los Diagramas de Clases de Análisis correspondientes al Modelo de Análisis. Este contiene todas las clases que se identificaron y componen el sistema junto con las relaciones entre clases, atributos y

operaciones.

**a) Servicio Ejecutar Práctica – Visualización de Componentes.**

La figura 3.2 muestra el diagrama de clases que participan en la implementación de la visualización de componentes, correspondiente al componente visual que da soporte al componente didáctico del laboratorio.



**Figura 3.2 Diagrama de clases para el Servicio Ejecutar Práctica – Visualización de Componentes**

El diagrama muestra una clase control que encapsula los mecanismos necesarios para realizar la presentación de las interfaces que se necesitan para esta práctica. Los estudiantes acceden a este componente a través de la interfaz gráfica que permite la selección de prácticas. Este componente está integrado al Laboratorio Virtual AXE-10 y permite la interacción directa con la Central AXE-10.

**b) Servicio Ejecutar Práctica – Gestión Servicios Especiales.**

La figura 3.3 muestra el diagrama de clases que participan en la implementación de la Gestión de Servicios Especiales, correspondiente al componente práctico que da soporte al componente “learning by doing” propiamente dicho.

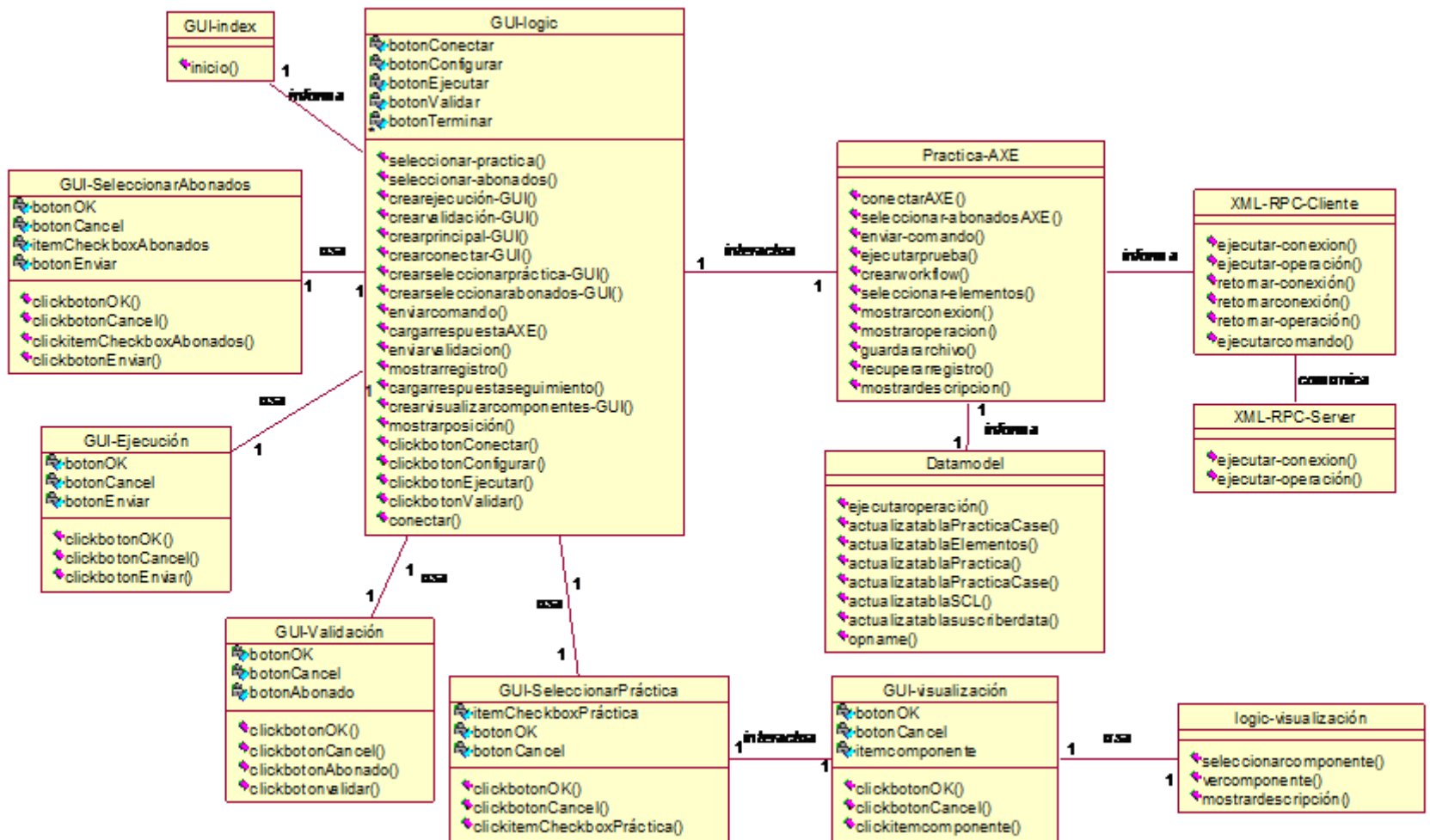
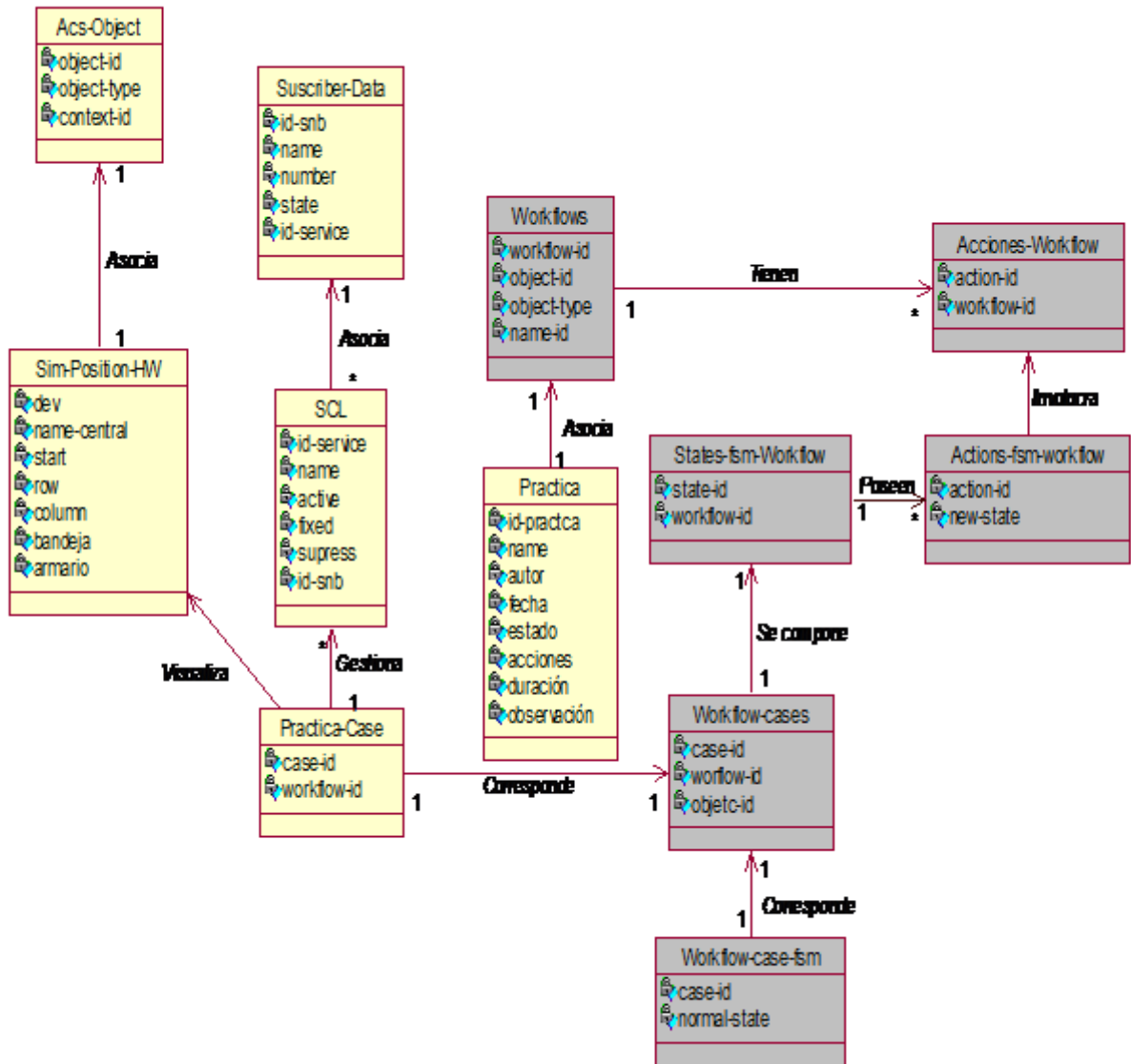


Figura 3.3 Diagrama de clases para el Servicio Ejecutar Práctica – Gestión de Servicios Especiales

El diagrama anterior permite entender las responsabilidades de las diferentes clases que participan, así como las relaciones que existen entre ellas. Como se puede observar, varias clases control encapsulan la lógica necesaria para realizar la conexión a la Central AXE-10, así como el control de los abonados que ya están en uso por otros estudiantes, el despacho de comandos y la recepción de los procesos que la Central retorna; este trabajo se realizará en estrecha colaboración con el modelo de datos, el cual utilizará clases “entity” que encapsulan la descripción de los objetos que participan en el desarrollo de una práctica virtual. Los estudiantes accederán a este componente a través de las interfaces gráficas diseñadas para entregar la práctica.

Para la entrega de una práctica y el seguimiento al desarrollo de las mismas, el componente práctico se apoya en la implementación de un modelo de datos de la figura 3.4, el cual guarda la información relativa a los abonados, las categorías, y demás objetos necesarios. Este modelo se encuentra enganchado al modelo de datos del paquete Workflow, con el fin de utilizar todas las capacidades que este servicio provee; lo cual se traduce en una entrega didáctica de la práctica a través de la secuencia de unos eventos previamente establecidos y que el estudiante debe seguir al interactuar con las interfaces. Además de hacerse una entrega didáctica de la práctica, este servicio permite el registro del desarrollo de las prácticas, así como las acciones que el estudiante realizó en el curso de la misma, eventos estos de gran ayuda para el profesor al momento de medir el progreso y desempeño.

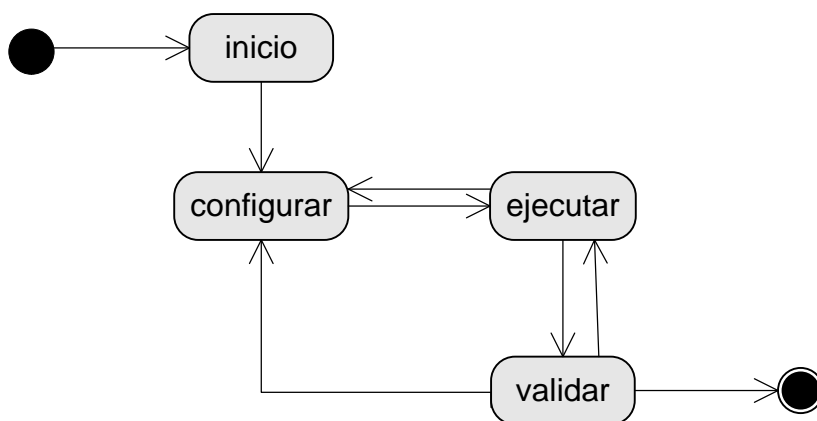


**Figura 3.4 Modelo de Datos para el Laboratorio Virtual AXE**

La definición de una máquina de estados finitos permite definir un workflow, el cual hace posible la entrega de la práctica de gestión de servicios especiales de una manera



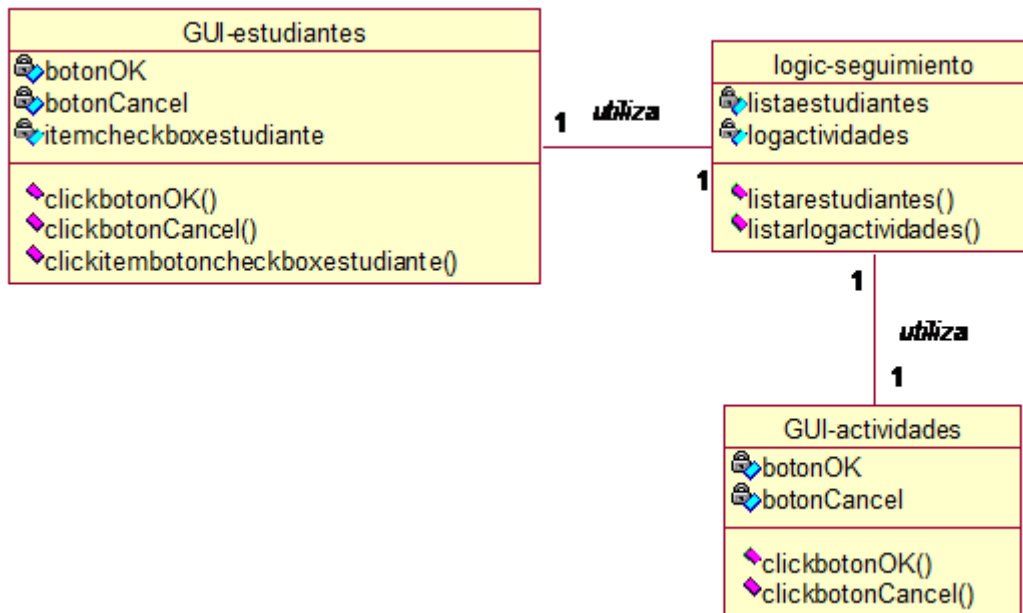
didáctica y controlada. Los estados del workflow definen unas tareas que el estudiante debe realizar para poder pasar a un estado próximo. Los diferentes estados del workflow se implementan en una interfaz principal, la cual guiará al estudiante a lo largo de la práctica. La implementación de esta máquina de estados pudo ignorarse, sin embargo debe recordarse que al diseñar un prototipo de laboratorio didáctico, debe implementarse un mecanismo que guíe a los estudiantes cuando están desarrollando la práctica y así se asegure el éxito del laboratorio, como el aprendizaje propuesto.



**Figura 3.5 Máquina de estados finitos para la entrega de la Práctica de Gestión de Servicios Especiales del Laboratorio Virtual AXE**

La figura 3.5 muestra la máquina de estados; en un primer estado, el estudiante inicia la práctica, evento en el cual el sistema se encarga de conectarse a la Central; posteriormente se pasa al estado configurar, donde el estudiante tiene que seleccionar los abonados; después de configurar el estudiante procede a ejecutar comandos. Del estado ejecutar puede regresar a configurar ó ir al próximo estado validar, en el cual probará las categorías gestionadas. Después de validar, el estudiante puede regresar a configurar otros abonados nuevamente, ó puede volver a ejecutar otros comandos nuevamente.

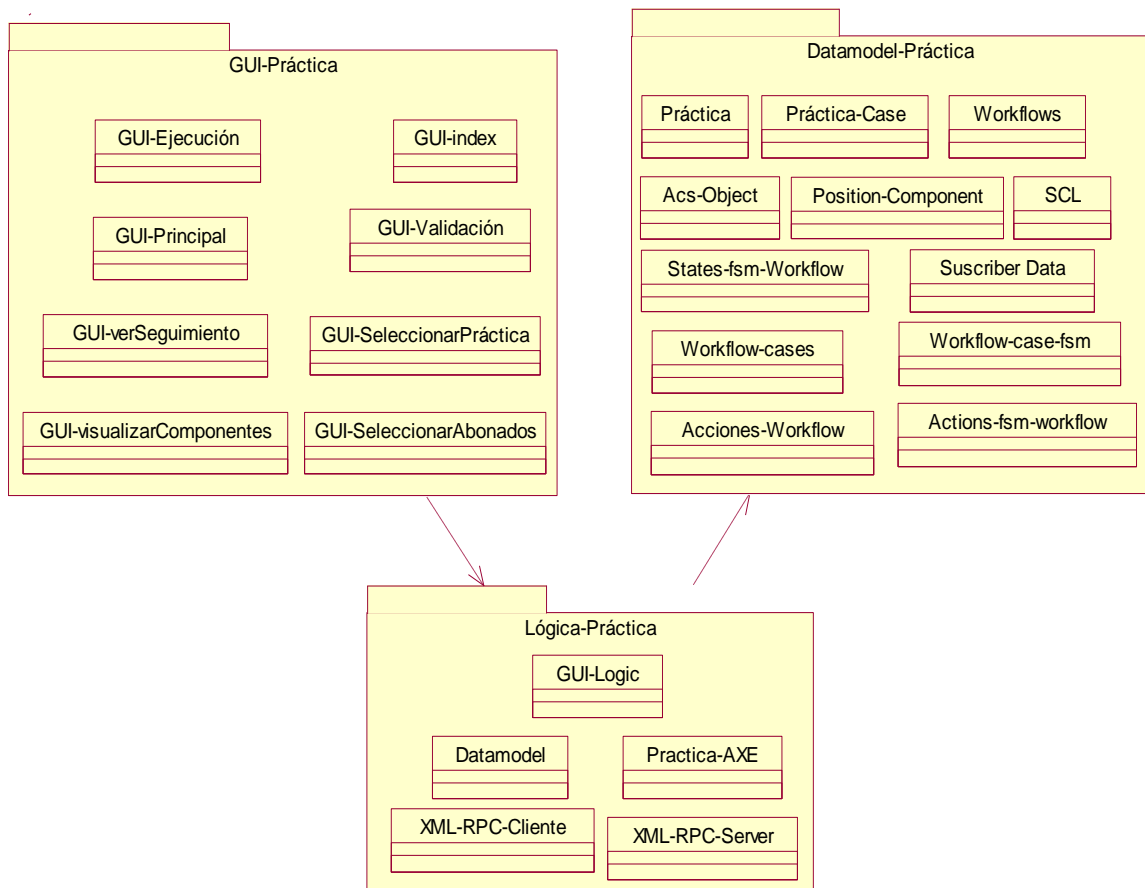
c) Servicio Ejecutar Práctica – Visualización de Seguimiento de Prácticas.



**Figura 3.6 Diagrama de clases para el Servicio Ejecutar Práctica – Visualización de Seguimiento**

La figura 3.6 muestra las clases que participan en la implementación del componente de seguimiento al desarrollo de las prácticas. En este diagrama se observa una clase “control”, encargada de hacer listado de los estudiantes y de las actividades que realizó en la práctica, para lo cual se ayuda de dos interfaces respectivamente.

### 3.3.2 Descripción de Paquetes y Clases de Análisis del Sistema.



**Figura 3.7 Paquetes de la Aplicación Laboratorio Virtual AXE**

Con base en el Modelo Esencial de Casos de Uso del Sistema se identificaron los

paquetes y clases evidentes arquitecturalmente significativos que se muestran en la figura 3.7. A partir de los casos de uso que se fijaron anteriormente se procedió a identificar las clases que implementan tales comportamientos del sistema.

**a) Descripción de las clases de análisis esenciales del paquete GUI-Práctica.**

Este paquete contiene archivos utilizados por la aplicación Laboratorio Virtual AXE-10, que aseguran las interfaces de usuario y su interacción con la aplicación. Entre las clases responsables de asegurar la interacción del usuario se muestran GUI-index, la cual constituye la interfaz inicial, y que da acceso a las diferentes opciones de prácticas, ya sean Visualización de Componentes ó Gestión de Servicios Especiales. La clase GUI-Selección-Práctica, se encarga de crear el formulario que mostrará las opciones principales del Laboratorio Virtual AXE-10, en este caso Visualización de Componentes o Gestión de Servicios Especiales. La práctica correspondiente a Gestión de Servicios Especiales presenta las clases GUI-Selección-Abonados, GUI-Ejecución, GUI-Validación.

**b) Descripción de las clases de análisis esenciales del paquete Lógica-Práctica.**

Este paquete contiene archivos, que aseguran el control del Laboratorio Virtual AXE-10 bajo el patrón modelo, vista, controlador. Las clases responsables de la lógica de la aplicación corresponden a GUI-Logic, Práctica-GUI-AXE, Práctica-GUI-Seguimiento. Por otra parte Datamodel es la clase responsable de la lógica asociada a los procesos de consulta, actualización e interacción con las tablas del modelo de datos.

EL paquete XML-RPC de OpenACS implementa un cliente simple XML-RPC. Cualquier paquete que necesite hacer llamadas XML-RPC puede simplemente agregar una dependencia a este paquete y luego llamar al método apropiado. XML-RPC-Server: Este servidor constituye una clase que es soportada por el entorno Java

### c) Descripción de las clases de análisis esenciales del paquete Datamodel-Práctica.

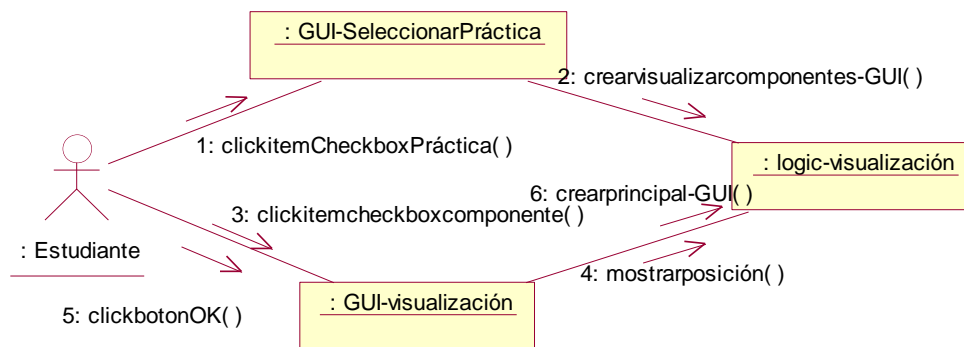
Este paquete almacena los diferentes objetos que conforman el modelo de datos necesarios para la ejecución de una práctica, así como toda la información correspondiente a los servicios que se prestan. Este modelo está integrado al modelo OpenACS de la plataforma. Entre las clases que aseguran el modelo de datos de la aplicación se encuentra, las clases Suscriber Data, y Práctica; las cuales constituyen el soporte para el modelo de datos. Este modelo de objetos que se implementan en tablas con unos atributos dinámicos, se integran a un modelo de workflow, el cual tiene estados, roles, tareas y acciones. La tabla Práctica-Case asocia una práctica con un workflow, es decir crea un caso de práctica del laboratorio. Una vez creada una práctica, ó en este caso un workflow, este tiene asociadas unas tareas, y unas acciones; las cuales se implementan bajo las tablas Actions-fsm-workflow, Acciones-Workflow, States-fsm-Workflow.

Como muestra el modelo de datos, cada workflow tiene asociado un objeto; un caso de práctica puede involucrar varios workflows; luego entonces cada workflow tiene unas tareas que se asocian a cada objeto, los cuales se relacionan en la tabla Object-map-task-prac. Por otra parte, la tabla Workflow-Cases, relaciona un objeto del modelo de datos, a un id-workflow, para generar un caso de workflow, que se especificará en otras tablas. Por su parte la tabla States-fsm-Workflow asocia un caso de práctica, con el estado en que se encuentra, ó estado normal.

Finalmente la tabla Workflows, contiene los objetos asociados a un workflow, así como el tipo de objeto, su descripción y la descripción del mismo workflow que contiene esos objetos.

### **3.3.3. Diagramas de Colaboración y/o Secuencia para los Casos de Uso del Sistema.**

**a) Servicio Ejecutar Práctica – Visualización de Componentes. (Componente Visual).**



**Figura 3.8 Diagrama de colaboración para el escenario Visualización de Componentes**

La figura 3.8 refleja un diagrama de colaboración para la implementación del escenario Visualización de Componentes, el cual comprende los casos de uso “Seleccionar Práctica” y “Ver componentes”, en el que se necesita la colaboración de varias clases. La figura 3.8 muestra clases tipo <<boundary>> encargadas de implementar las interfaces para visualización de los componentes, mientras que la clase <<control>> implementan la lógica que permite la selección y descripción del componente.

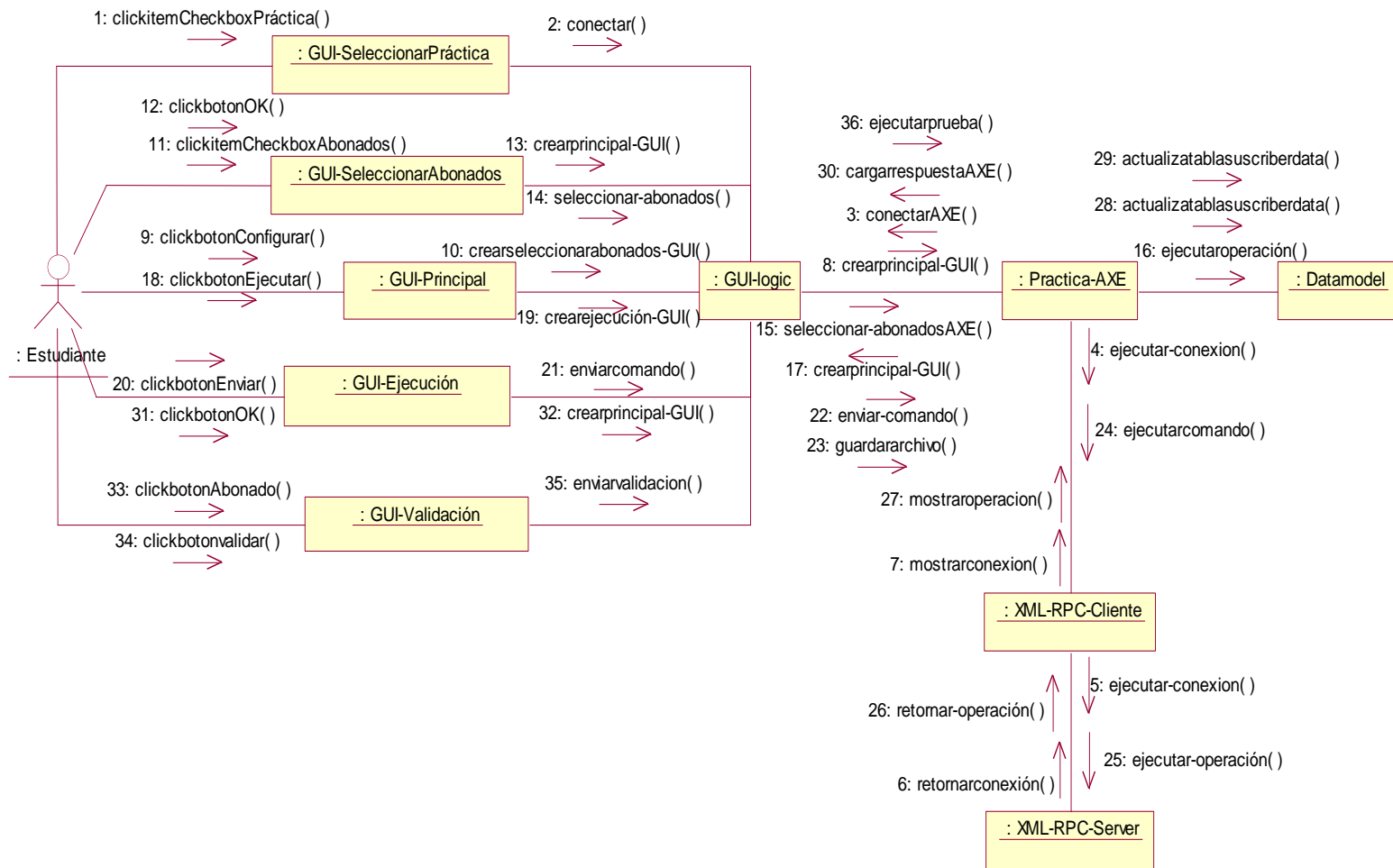
**b) Servicio Ejecutar Práctica – Gestión de Servicios Especiales. (Componente Práctico).**

La figura 3.9 refiere un diagrama de colaboración para la implementación del escenario gestión de servicios especiales, el cual involucra los casos de uso “Seleccionar-Práctica” y “Gestión de Servicios Especiales”. Este diagrama muestra que se necesitan varias clases <<boundary>>, las cuales modelan interacción entre el laboratorio y los estudiantes ó profesores e implementan las interfaces gráficas de Selección de Práctica, Ejecución, Validación, así como la interfaz principal, responsable de fijar los estados de desarrollo de una práctica. También se requiere de la participación de tres clases <<control>> las cuales encapsulan la coordinación y secuenciamiento necesario para implementar la lógica de desarrollo de las prácticas,

así como ejecutar las diversas operaciones sobre el modelo de datos.

Por otra parte, se necesitan algunas clases <<entity>> que modelan la información persistente que se necesita para el desarrollo de las prácticas, tales clases se implementan en las tablas correspondientes del modelo de datos.

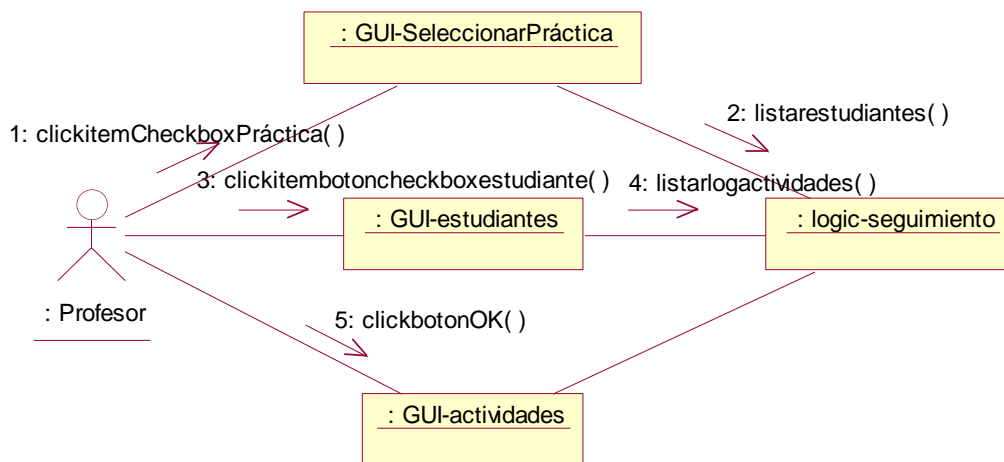
Para asegurar la correcta transmisión de los comandos hacia la Central AXE-10, se necesita la colaboración de varias clases, las cuales son las misma que participan en la conexión; este comportamiento se logra con las clases XML-RPC Server y XML-RPC cliente.



**Figura 3.9 Diagrama de colaboración para el escenario Gestión de Servicios Especiales**



c) Servicio Ejecutar Práctica – Ver Seguimiento (Componente Didáctico).



**Figura 3.10 Diagrama de colaboración para el caso de uso Ver Seguimiento**

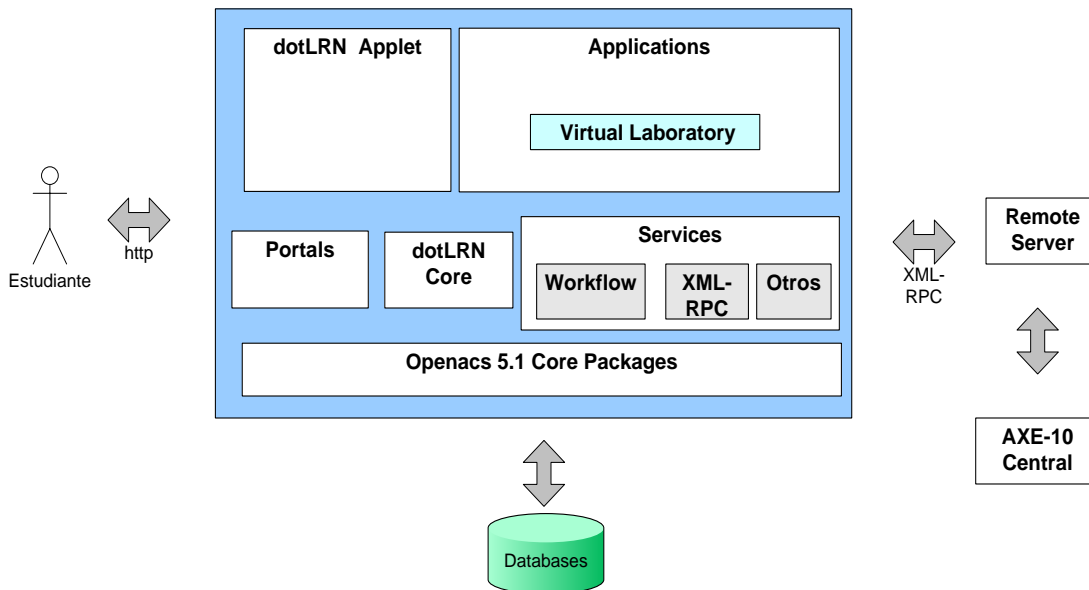
La figura 3.10 muestra el componente didáctico del Laboratorio virtual AXE-10, el cual se encarga de registrar los eventos y las acciones realizadas por los estudiantes en las prácticas. Las clases que participan de este escenario, están conformadas por GUI-seleccionarPráctica, la cual permite el ingreso del estudiante a este escenario, así como GUI-estudiantes, y GUI-actividades encargadas de presentar la selección de los estudiantes y mostrar los estados de las prácticas desarrolladas. La lógica de este escenario está bajo responsabilidad de la clase logic-seguimiento responsable de crear las anteriores interfaces.

### 3.4 MODELO DE DISEÑO DEL SISTEMA.

#### 3.4.1 Arquitectura de Referencia para el Sistema.

Al inicio del capítulo se ubicó el Laboratorio Virtual AXE-10 dentro de una de las dos grades categorías. En este orden de ideas, el Laboratorio Virtual AXE-10 se ubica dentro de la categoría de aquellos laboratorios que hacen procesamiento en el servidor, por cuanto la aplicación desarrollada está corriendo sobre el servidor y es accedido por los estudiantes remotamente a través de un browser, el cual es usado para entrar comandos y presentar sus resultados en una forma agradable.

La arquitectura de referencia para la aplicación del paradigma learning by doing mediante la construcción de laboratorios virtuales en la plataforma dotLRN del proyecto E-LANE, se fundamenta en la arquitectura que se expuso en el capítulo dos.



**Figura 3.11 Arquitectura Distribuida para el Laboratorio Virtual AXE-10**

La figura 3.11 muestra los principales subsistemas que interactúan para cumplir con la funcionalidad del Laboratorio Virtual AXE-10, de la cual forma parte OpenACS con su LMS dotLRN, un servidor remoto, y el sistema real al cual se desea acceder, que para este caso es la Central AXE-10.

### **3.4.2 Descripción de Subsistemas.**

El sistema real al cual el estudiante tiene acceso, es la Central AXE-10, y se entiende como las tres letras que corresponden a un código de un producto de Ericsson, los números indican la variante del producto. El sistema AXE es un sistema “Stored Program Control” (SPC), lo cual significa que programas almacenados en un computador controlan la operación de la Central. El sistema AXE está formado por dos partes principales: el equipo de conmutación para la conexión de la llamada telefónica y un computador para controlar el equipo de conmutación. El equipo de conmutación es llamado APT, en otras palabras es la parte telefónica. La APZ, es la parte de control, consiste de Hardware, y el software para la gestión de los reportes y para la administración del trabajo que hace el computador.

Por otra parte dotLRN, es un subsistema que ya se describió en el capítulo dos, sin embargo debe recordarse que es el subsistema, al cual el Laboratorio Virtual AXE-10 está integrado. Se considera el LMS de OpenACS, que ofrece varias aplicaciones colaborativas como son calendarios, foros, ficheros compartidos, etc.; además de los que ofrece OpenACS. Este subsistema se comunica con la Central AXE-10 a través del servidor Remoto.

El subsistema Workflow, corresponde a un servicio de OpenACS, el cual agrega capacidades útiles a dotLRN, al usarse para hacer aplicaciones de una forma estandarizada. Los workflow habilitan a las aplicaciones para usar todas las alertas y las capacidades de gestión de eventos construidas en el paquete Workflow. El sistema

Workflow podría ejecutar determinada actividad cuando una tarea ha sido terminada o se necesita empezar, podría poner en orden las tareas para mostrar a los usuarios.

Por otra parte XML-RPC, es un protocolo que usa XML sobre http y que permite a los clientes hacer llamadas a procedimientos remotos sobre su servidor. El dato es transformado dentro de un formato XML estándar antes de ser transferido entre el cliente y el servidor. Esto permite a lenguajes de programación y sistemas operativos diferentes ínteroperar. Este paquete implementa un cliente simple XML-RPC. Cualquier paquete que necesite hacer llamadas XML-RPC puede simplemente agregar una dependencia a este paquete y luego llamar al método apropiado.

Otros subsistemas, corresponden a otros servicios y aplicaciones que un laboratorio virtual puede utilizar al estar integrado en un LMS, los cuales ya se describieron en el capítulo dos (2).

### **3.4.3 Descripción de las Clases de Diseño.**

Desde el punto de vista de la arquitectura propuesta y en el marco del proceso de desarrollo de la aplicación para la validación de la misma, es necesario describir las clases de diseño que explican el patrón de diseño Modelo – Vista – Controlador, así como las clases de los subsistemas que permiten la integración de la aplicación, la plataforma dotLRN a la Central AXE -10, subsistemas que componen el Laboratorio Virtual AXE-10.

Las clases correspondientes al subsistema “Presentación” involucran todas las interfaces gráficas de usuario con las cuales el estudiante interactúa, y se corresponden a los diversos estados en los que una práctica se puede encontrar. Las clases que

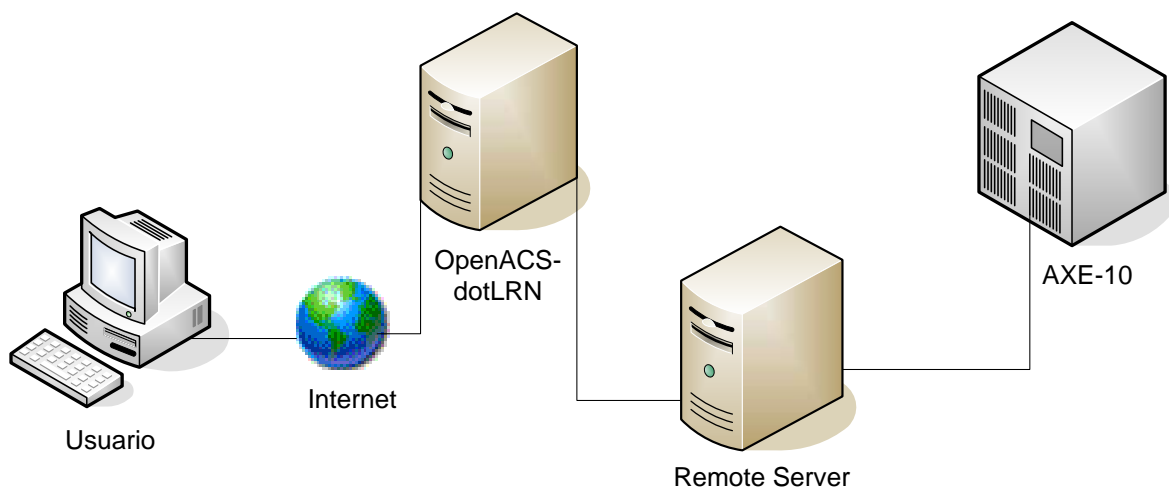
facilitan el comportamiento de la Lógica se encargan de todos los procesos asociados a la funcionalidad necesaria para la interacción entre el usuario, la plataforma y la Central AXE. Por tanto las clases tipo controlador aceptan los eventos de entrada, traduce los eventos de entrada a peticiones que se hacen al modelo o a las vistas, e implementan los procedimientos necesarios para actualizar si es necesario.

Por su parte el Modelo de Datos, permite la persistencia de la información de las prácticas que se realicen en el Laboratorio Virtual AXE-10, la cual está soportada por un modelo de datos que se integra al modelo de OpenACS. La integración de estos modelos de datos no solo permite reusabilidad, sino la integración del Laboratorio Virtual AXE-10 a servicios de OpenACS tales como workflow, lo cual se traduce en una alta funcionalidad en términos de entrega del laboratorio y seguimiento.

En términos de comunicación el Cliente XML-RPC permite hacer llamadas XML-RPC a procedimientos remotos, para lo cual simplemente se debe agregar una dependencia al paquete apropiado y luego llamar al método respectivo. Finalmente el Servidor XML – RPC constituye una clase que es soportada por el entorno JAVA, y que atiende las peticiones del cliente XML-RPC.

### **3.5. MODELO DE DESPLIEGUE DEL SISTEMA**

La arquitectura para la aplicación del paradigma learning by doing mediante la construcción de laboratorios virtuales en la plataforma dotLRN se muestra en la figura 3.12; básicamente se compone de dos nodos principales la plataforma OpenACS-dotLRN, y un servidor remoto que se comunicará con la Central Ericsson AXE-10.



**Figura 3.12 Arquitectura Básica**

La comunicación entre el cliente y la plataforma OpenACS se realiza a través del protocolo http, mientras que la comunicación entre la plataforma y el servidor remoto se implementará vía XML-RPC.

Los estudiantes pueden acceder al laboratorio usando un browser WWW, el cual presenta una página ADP proveída por el servidor AOLserver, esta comunicación es mediada por la plataforma dotLRN y será un proceso transparente. Este documento contendrá los elementos necesarios para iniciar y detener las aplicaciones proveídas por el servidor vía Internet; de tal forma que ningún software adicional es requerido aparte del browser en si mismo.

La funcionalidad del lado de la plataforma dotLRN es implementada por el Laboratorio Virtual AXE-10 que se encuentra integrad a la misma e interactúa con los servicios básicos de los paquetes Workflow y XML-RPC. La lógica de esta aplicación se resume en la interacción de una práctica con el subsistema Workflow, el cual permite hacer el seguimiento de los diversos estados en que se puede encontrar esa práctica y por lo tanto el proceso de aprendizaje del estudiante.

El servidor WWW cuya función es proveer documentos adps embebidos de elementos e-learning, los cuales corresponde a las interfaces necesarias para la ejecución de las

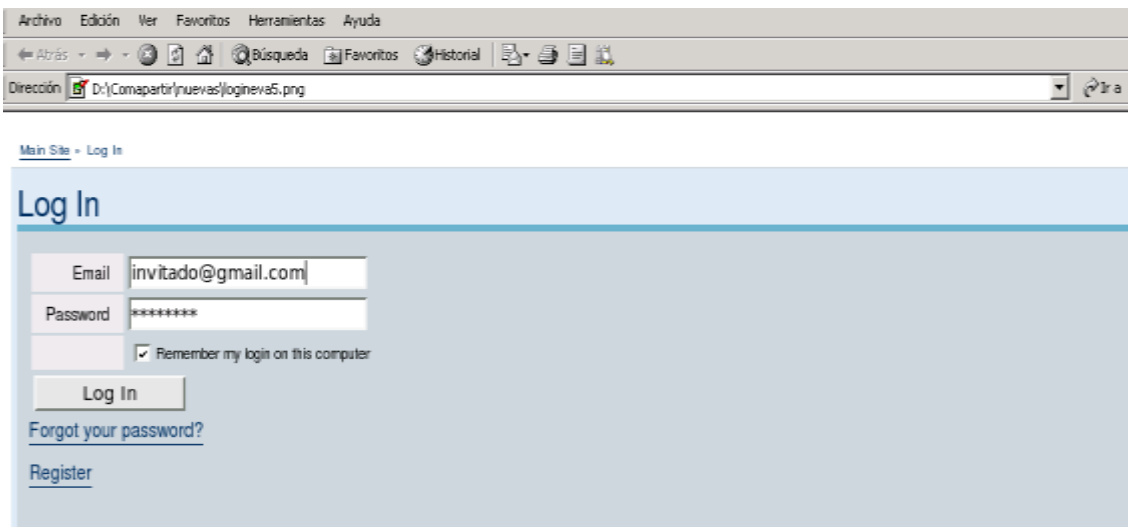
prácticas establecidas como objetos de aprendizaje.

El servidor XML –RPC soportado por el entorno Java, se encarga de recibir las peticiones realizadas por la plataforma dotLRN, procedentes de los comandos entrados por el usuario y a su vez entregarlos a la Central AXE-10; la cual realiza los debidos procesos, que retornará a este servidor. Resultados que posteriormente serán mostrados al usuario, a través de las respectivas interfaces de la plataforma dotLRN.

Finalmente, la plataforma se encarga de soportar los servicios funcionales agregados del Laboratorio Virtual AXE-10, tales como: autenticación de usuario, almacenamiento de datos de usuario, herramientas de comunicación y otros que adicionan un alto valor pedagógico al sistema.

### **3.6 INTERFACES GRAFICAS DEL PROTOTIPO FINAL**

El objetivo de este apartado es describir las interfaces que se le presenta al estudiante y al profesor para interactuar con los componentes básicos que el sistema provee. A continuación se presentan las interfaces que el sistema muestra al estudiante en el desarrollo de las prácticas realizadas en el laboratorio virtual AXE-10. La primera interfaz que la plataforma presenta al estudiante corresponde a la interfaz de ingreso donde el estudiante debe introducir el correo y el password asociado al mismo para poder ingresar a la plataforma.

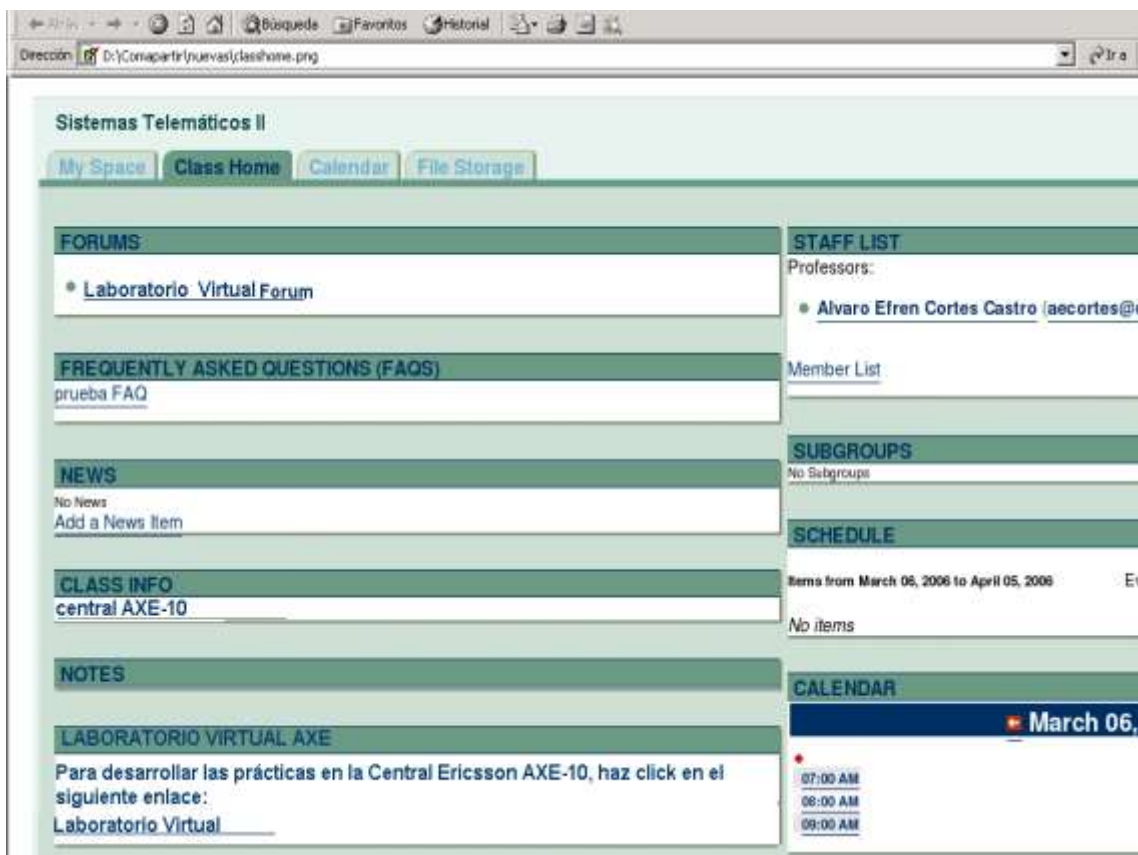


**Figura 3.13 Interfaz de validación de ingreso**

Después de haber ingresado a la plataforma el sistema le presenta la página de inicio de cada usuario (home), dependiendo de los privilegios que tenga (profesor, estudiante); en este caso como es un estudiante se le presenta “My Space”, página donde residen las clases en las cuales está inscrito y desde donde puede acceder a ellas.

En la pestaña del Class Home, (Figura 3.14) correspondiente a la materia Sistemas Telemáticos II, residen diferentes portlets asociados con los diferentes servicios que el LMS puede proveer al curso, tales como foro, calendario, notas, staff, subgrupos y otros, cuya importancia y necesidad se explicó en el capítulo II; también se encuentra el portlet que contiene el Laboratorio Virtual AXE-10, el cual le permite al estudiante la realización de las prácticas básicas del curso de Laboratorio II de Sistemas Telemáticos.





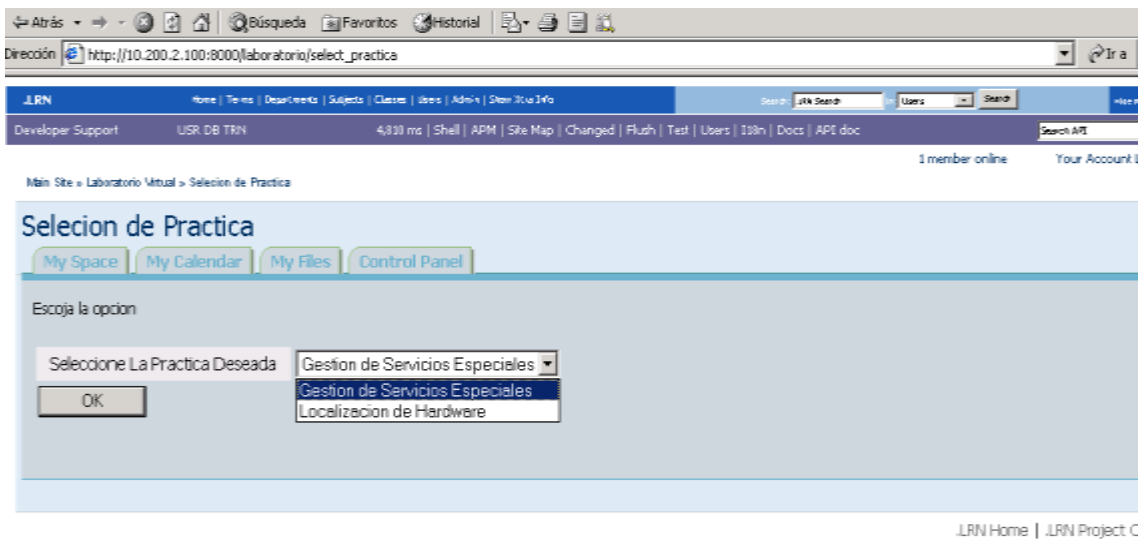
**Figura 3.14 Interfaz Class Home**

Después de haber dado clic en el enlace correspondiente al laboratorio virtual del portlet del class Home, el sistema le presenta la interfaz que invita al estudiante a entrar a la aplicación, la cual presenta una breve descripción del laboratorio – figura 3.15.



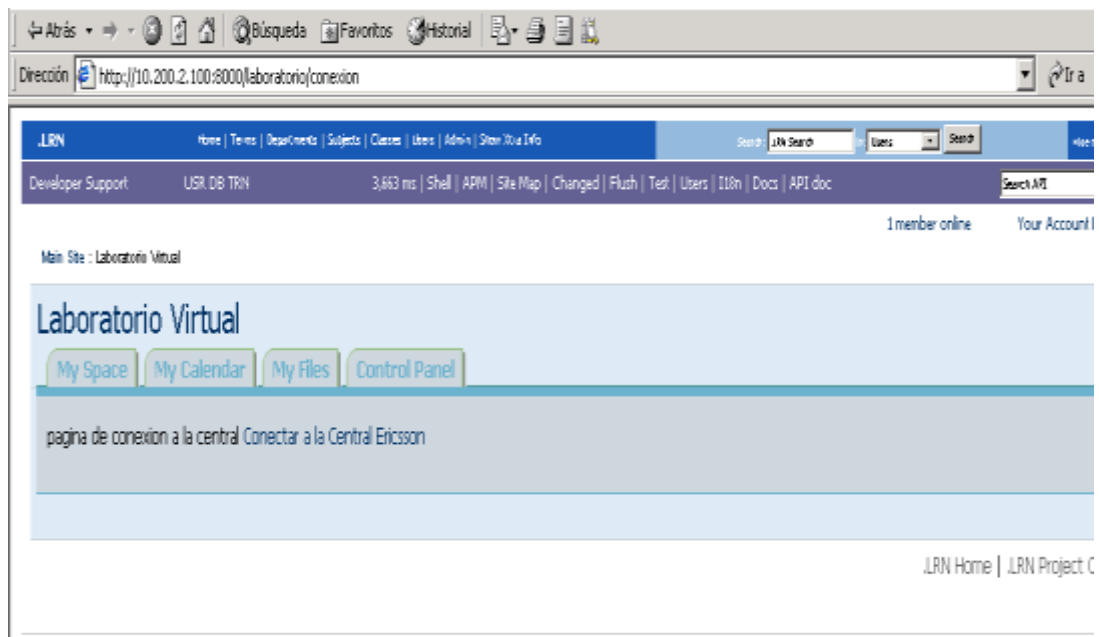
**Figura 3.15 Interfaz de Presentación**

Después de haber ingresado a la aplicación, el sistema presenta al estudiante la interfaz de Selección de práctica, que le permite seleccionar uno de los dos componentes del laboratorio, ya sea la visualización de componentes ó la gestión de servicios especiales (figura 3.16).



**Figura 3.16 Interfaz de Selección de Práctica**

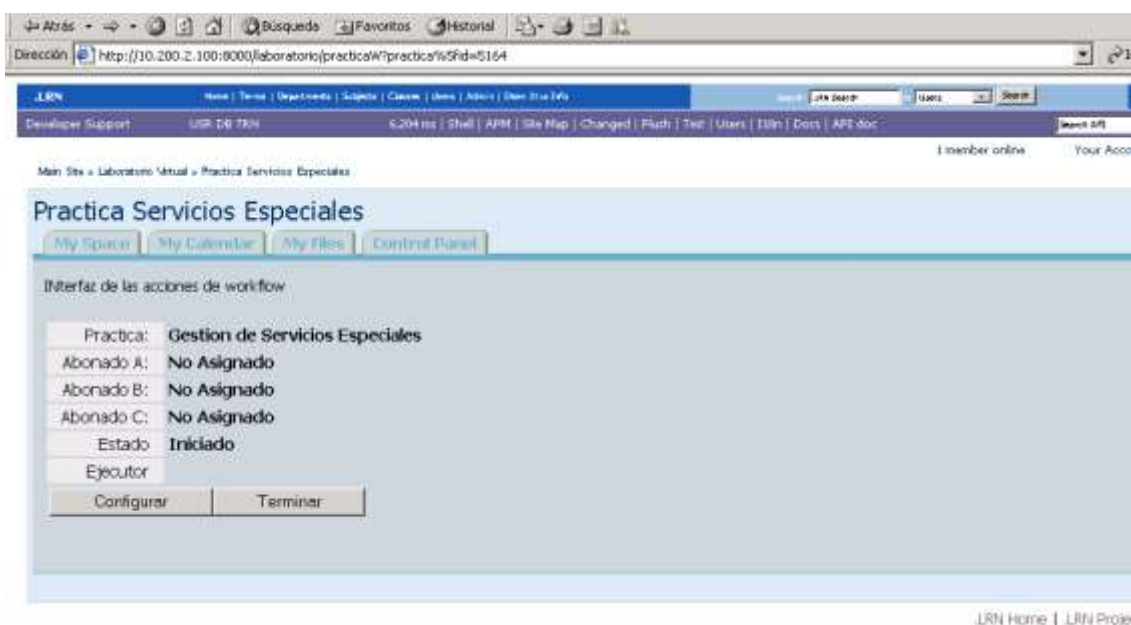
Después de haber seleccionado la opción correspondiente a la Gestión de Servicios Especiales, el sistema le presenta la interfaz que le permite conectarse con la Central Ericsson AXE-10 (figura 3.17).



**Figura 3.17 Interfaz de Conexión del Laboratorio Virtual**

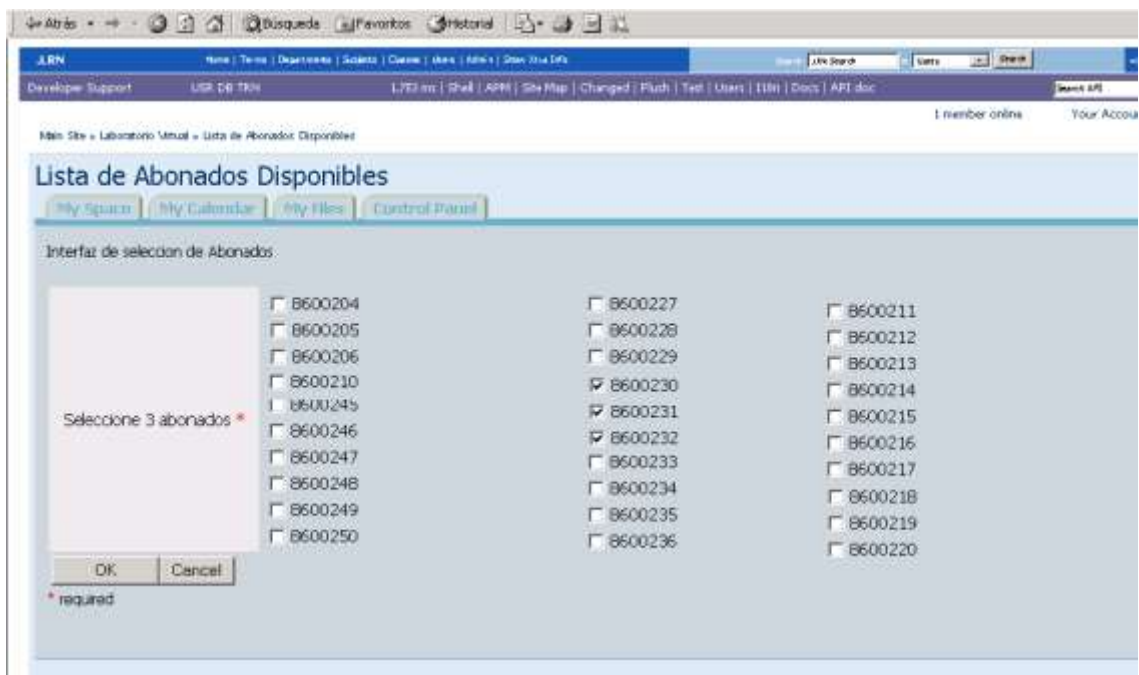
Después de una conexión exitosa, el sistema le presenta la Interfaz Principal correspondiente a la práctica de servicios especiales, que en adelante será el mecanismo de entrega de la práctica. En caso contrario se mostrará una interfaz de error.

Esta interfaz principal es la representación de la máquina de estados finitos, es decir, a medida que el estudiante vaya realizando las diferentes etapas del laboratorio, habilita o deshabilita unas determinadas acciones. Cuando el estudiante empieza una práctica, la interfaz se encuentra en su estado iniciado, solo le permite al estudiante configurar los abonados o dar por terminado la práctica (figura 3.18).



**Figura 3.18 Interfaz Principal de Práctica de Servicios Especiales**

Si el estudiante decide configurar los abonados con los cuales va a trabajar en la práctica, el sistema le presenta la interfaz correspondiente a la Lista de Abonados Disponibles, de los cuales sólo podrá seleccionar tres. Una vez los abonados se han seleccionado, la interfaz no permitirá su selección por otro usuario, evitando así posibles interferencias que se pudieren presentar (figura 3.19).



**Figura 3.19 Interfaz de Selección de Abonados Disponibles**

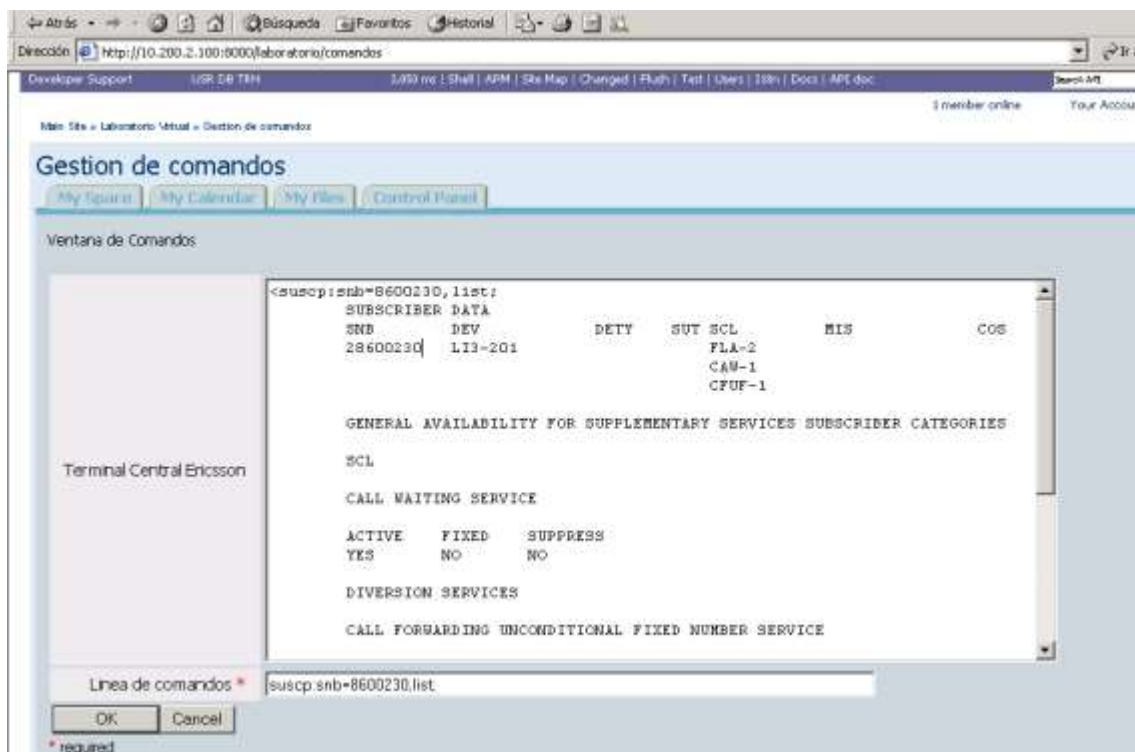
Después de haber seleccionado los tres abonados, el sistema le muestra nuevamente la Interfaz Principal, la cual presenta el nombre de la práctica, los abonados seleccionados, el estado de la práctica, y los botones que le permiten al estudiante regresar al estado anterior (configurar) ó ir a un estado próximo (ejecutar) y terminar (figura 3.20).



**Figura 3.20 Interfaz Principal de Práctica de Servicios Especiales**

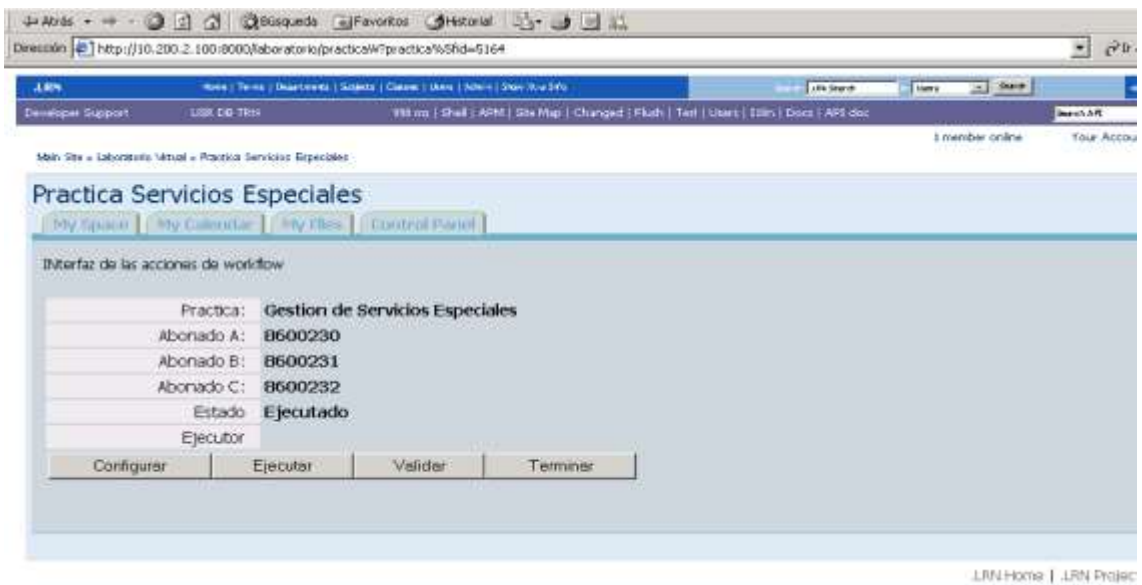
Si el estudiante selecciona el botón ejecutar, el sistema le muestra una interfaz de ejecución, la cual básicamente contiene la ventana para el ingreso de los comandos que se enviarán a la central y otra ventana para recibir la respuesta retornada por la misma (figura 3.21).

La interfaz de ejecución le permite al estudiante ingresar los comandos según la sintaxis establecida. Una vez enviado un comando, la respuesta retornada por la central es mostrada en una caja de texto.



**Figura 3.21 Interfaz de Ejecución**

Después de haber interactuado con la Central Ericsson AXE-10, el estudiante retorna a la Interfaz Principal, la cual le presenta de nuevo los abonados con los que está trabajando y el estado de la práctica, así como los botones correspondientes a los estados anteriores a los cuales puede regresar y el estado próximo al cual puede ir (Validar), ver figura 3.22.



**Figura 3.22 Interfaz Principal de Práctica de Servicios Especiales**

Una vez que el estudiante ha asignado un servicio especial a un abonado a través de comandos, podrá probarlo de manera simulada mediante una interfaz que el sistema le provee a través del botón validar (figura 3.23).



**Figura 3.23 Interfaz de Validación**

En esta interfaz se le presenta los tres abonados que el estudiante seleccionó para



trabajar, que al igual que en la práctica real sirven para poder verificar los servicios especiales habilitados ya que estas unidades tienen características y funciones similares tales como digitar un número, timbrar, colgar, descolgar, etc.

En todas las interfaces existe un botón terminar; en el momento que el estudiante desee terminar la práctica, oprime este botón dejando los tres abonados libres para que otro estudiante los pueda tomar, además de llevarlo nuevamente a la interfaz de selección de práctica (figura 3.16).

Una vez el estudiante ha seleccionado la opción componente (figura 3.16), el sistema le presenta una interfaz que le presenta una vista frontal y posterior de la Central AXE-10, las cuales le permiten al estudiante conocer los componentes dentro de la ubicación física del sistema y conocer una descripción del mismo (figura 3.24).



**Figura 3.24 Interfaz de Visualización de Componentes**

Cuando el estudiante selecciona uno de los componentes de la lista de la izquierda, se activa un archivo de video y de audio que le presenta una descripción funcional del mismo.



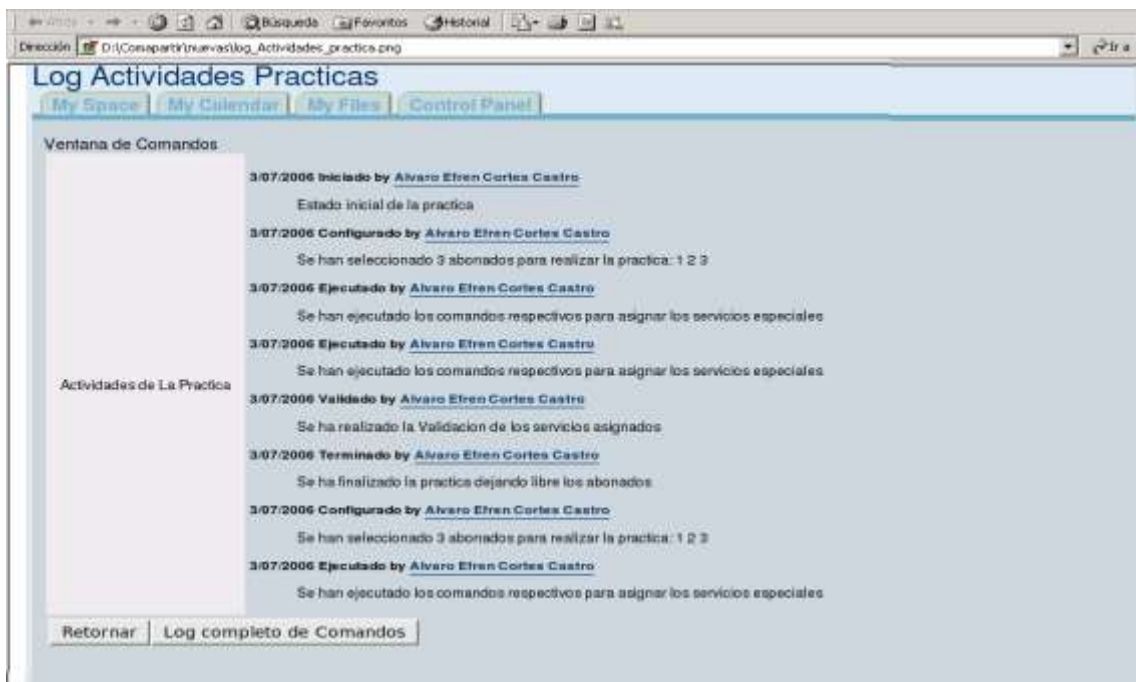
**Figura 3.25 Interfaz de Visualización de Componentes**

Las anteriores interfaces representan el componente práctico y visual que está dirigido a los estudiantes. En cuanto al profesor, la aplicación le ofrece la posibilidad de hacer seguimiento a las prácticas que el estudiante ha realizado, a través de una interfaz que le permite seleccionar los estudiantes y detallar las prácticas realizadas en el laboratorio (figura 3.26).



**Figura 3.26 Interfaz de Seguimiento de Prácticas**

La selección de un estudiante, permite detallar el desarrollo de las prácticas en sus diversos estados, así como la fecha en que se realizó tales prácticas (figura 3.27). El conocimiento de tales estados ofrece la posibilidad de conocer el progreso y los posibles inconvenientes que se puedan presentar en el proceso de aprendizaje.



**Figura 3.27 Interfaz de Actividades de Prácticas**

Finalmente debe mencionarse que el profesor puede detallar todas las acciones que el estudiante realizó al interactuar con la central AXE-10, debido a que todos los comandos enviados a ésta, así como sus respuestas son guardadas en la base de datos.

#### **IV. CONCLUSIONES Y RECOMENDACIONES**

La aplicación del Paradigma Learning by Doing mediante la construcción de laboratorios virtuales en el proyecto E-LANE, en la Universidad del Cauca, constituye la primera aproximación hacia una revolución educativa, enmarcada en los nuevos cambios que exige la sociedad en términos de demandas educativas y sociales. El trabajo permitió inferir algunas conclusiones y avances en el campo tecnológico, así como en el campo teórico.

El trabajo permitió realizar una consulta bibliográfica amplia que resume el estado del arte del aprendizaje práctico, cuyo análisis realizado establece una separación entre los sistemas que entregan recursos educativos orientados al aprendizaje teórico (principalmente expositivos) y aquellos orientados al aprendizaje práctico (principalmente activos). El desarrollo de los segundos se ha realizado más tarde que los primeros debido a una superior dificultad para su desarrollo y a unos requisitos tecnológicos mayores. Precisamente por ello, el número de sistemas que implementan el paradigma del laboratorio virtual en entornos de aprendizaje electrónico a distancia es muy inferior al de aquellos que ofrecen entornos de aulas virtuales.

En forma paralela el trabajo permitió fijar la atención en última instancia en los laboratorios virtuales como aplicaciones e-learning, con el propósito de conocer sus principales elementos y sus características, las cuales redundarán en una unidad de aprendizaje de mayor eficiencia al integrarse a un LMS, para lo cual se pueden establecer los criterios que describen el contenido de un laboratorio, así como los elementos para su distribución.

El trabajo realizado permitió establecer la caracterización de los componentes básicos que forman un laboratorio virtual al estar integrado en un LMS, los cuales se ubican en el campo de la gestión y la distribución; componentes que son entregados a través del equivalente de una “solución stack” consistente de tres capas funcionales: una capa de

aplicación, una capa “plataforma”, y la infraestructura virtual.

El diseño e implementación del prototipo Laboratorio Virtual AXE-10 para la materia de Sistemas Telemáticos II integrado al LMS dotLRN, no sólo permitió la validación de la arquitectura propuesta del paradigma práctico, sino conocer de manera general las ventajas y desventajas de OpenACS y su LMS.

En cuanto a ventajas de ésta tecnología se pudo establecer sus capacidades y funcionalidades en el desarrollo de aplicaciones Web orientadas a comunidades, así como sus perspectivas en términos de desarrollo de nuevas aplicaciones para e-learning; específicamente se demuestra que la construcción de laboratorios virtuales en el marco del proyecto E-LANE, en la Universidad del Cauca debe aprovechar los servicios que el LMS ofrece, así como otras aplicaciones y servicios tales como workflow y XML-RPC, los cuales permiten la integración de sistemas reales y virtuales, así como la entrega didáctica de los mismos.

Por otra parte, se evidenció las capacidades didácticas de laboratorios virtuales, dado que al ser aplicaciones que modelan un sistema, permiten a los estudiantes manipular variables de entrada para cambiar el comportamiento del sistema y ver los resultados.

En cuanto a desventajas se pudo establecer las dificultades que se presentan al desarrollar nuevas aplicaciones, dado que la documentación es escasa; lo cual lleva a que la curva de aprendizaje sea lenta; situación que se deriva de la naturaleza de la plataforma, al ser software libre.

Las nuevas demandas sociales han producido un cambio en los paradigmas educativos, frente a las cuales Internet juega un papel esencial. Tales demandas sociales imponen modelos educativos centrados en el alumno, donde se busca una formación continua y “de por vida”. Se trata de formar profesionales con poco tiempo disponible y con un elevado rendimiento de su tiempo.

La apropiación de una tecnología, aunque requiere esfuerzo e investigación, puede constituirse en una opción válida para dar solución a situaciones concretas de una región.

## **Recomendaciones**

En cuanto a documentación, el desarrollador debe apoyarse en las fuentes que proporciona el sitio oficial de OpenACS, puesto que allí se encuentra la mayor documentación existente, tanto relativa al desarrollo de aplicaciones de OpenACS, como referidas al manejo de los lenguajes de programación que esta tecnología usa; sin embargo ésta no es suficiente al momento de buscar un tema específico, por lo cual se recomienda explorar otros paquetes que pueden proporcionar valiosa información.

Una fuente de valiosa información lo constituyen los foros; en este sentido cuando se vaya a realizar una consulta, por experiencia, se recomienda realizarlas lo más puntual posible, con el objetivo de obtener una respuesta satisfactoria.

Finalmente para facilitar el proceso de desarrollo en OpenACS se recomienda un adecuado manejo en bases de datos como Oracle, Postgres, Tcl, y Linux.

En cuanto al desarrollo de un laboratorio virtual se pudieron deducir las siguientes recomendaciones:

El desarrollo de laboratorios virtuales debe involucrar un equipo interdisciplinario. Para el análisis del dominio, es necesario las bases teóricas, y un conocimiento comprensivo acerca del laboratorio y los procedimientos que se realizan en él.

Un laboratorio virtual debe incorporar un concepto didáctico, el cual describe la lógica de las prácticas, mediante una estructura de navegación. Puede ayudarse de

visualización de las relaciones, establecer momentos para un aprendizaje reflexivo. Así como también debe presentar un panorama general del desarrollo de la práctica, con sus diferentes etapas.

### **Trabajos Futuros.**

Con el ánimo de dar continuidad a la “aplicación del paradigma learning by doing mediante la construcción de laboratorios virtuales al proyecto E-LANE en la Universidad del Cauca”, se recomienda el planteamiento de los siguientes posibles proyectos como trabajos futuros de desarrollo e investigación en ésta área:

**Construir diversos prototipos correspondientes a las clasificaciones del paradigma learning by doing expuestas en el estado del arte.** El prototipo diseñado e implementado permitió explorar las capacidades de la plataforma para integrar sistemas de acceso real; sin embargo debe plantearse la construcción de otros prototipos que se ubiquen dentro de las otras clasificaciones expuestas en el estado del arte, tales como laboratorios virtuales propiamente dichos, ya sean ejecutados en el cliente ó en el servidor. Esto permitirá un amplio conocimiento de OpenACS y su LMS. La construcción de tales prototipos pueden ser de otras disciplinas y debe hacerse no solo en conjunto con una exploración tecnológica de otros servicios y aplicaciones de la plataforma, que sumen un mayor valor agregado, sino con la ejecución de un trabajo, que involucre personas de las diferentes disciplinas del ámbito educativo.

**Desarrollo de una metodología para el diseño de laboratorios virtuales.** El desarrollo de una metodología para el diseño y construcción de laboratorios virtuales, debe partir de la exploración tecnológica profunda de la plataforma, la experiencia adquirida en el diseño de los prototipos ajustados a las demandas de la región y el país, y un trabajo interdisciplinario que soporte los objetivos educativos que se persiguen.



## V. REFERENCIAS BIBLIOGRAFICAS

- [SC02] Serrano Castaño, Carlos Enrique. 2002. *Un Modelo Integral Para Un Profesional En Ingeniería*. Editorial Universidad del Cauca, Popayán.
- [GP04] García Pozo Pedro J. 2004. *Implantación de dotLRN en la Universidad de Valencia. Proyecto Aula Virtual*. Consultada en: <http://www.uv.es/ticape/docs/pedroj/proyecto.pdf>.
- [AC02] Aldrich Clark. 2002. *Simulations and the future of learning. An innovative and perhaps revolutionary approach to e-learning*. Pg. 240.
- [AF01] Anido Luís, Llamas Martín and Fernández J. Manuel. 2001. *Internet-based Learning by Doing*. Consultada en: <http://www.ewh.ieee.org/soc/es/May2001/17/Begin.htm#fig2>.
- [CG01] Castellanos Germán, Giraldo Aldemar. 2001. *E-Lab*. Consultada en: Primer Congreso Iberoamericano de telemática. Memorias. Cartagena de Indias, Agosto.
- [JDS00] Spanias Andreas, Constantinou Argyris, Foutz Jeff, Bizuneh Fikre. 2000. *An online signal processing laboratory*. Consultada en: <http://spib.ece.rice.edu/DSP2000/submission/SPE/papers/paper028/paper028.pdf>
- [SU00] Spanias Andreas, Urban Susan, Constantinou Argyris, Tampi Maya, Clausen Axel, Zhang Xiaopeng, Foutz Jeff and Stylianou Georgios. 2000. *Development and evaluation of a web-based signal and speech processing laboratory for distance learning*. Consultada en: <http://www.eas.asu.edu/~spanias/icassp2000.pdf>

- [AF03] Anido Luís. Llamas Martín and Fernández J. Manuel. 2003 *Experiencias en el uso del sistema de laboratorios virtuales Simulnet*. Consultada en: [http://www.quadernsdigitals.net/datos\\_web/hemeroteca/r\\_14/nr\\_197/a\\_2795/2795.html](http://www.quadernsdigitals.net/datos_web/hemeroteca/r_14/nr_197/a_2795/2795.html).
- [Mat] *Mathematica as a "front-end" to External Programs*. Consultada en: <http://www.ewh.ieee.org/soc/es/Nov1998/13/EXAMPLEC.HTM>.
- [E-lane] <http://e-lane.org>.
- [CE03] Calvo Rafael. Ellis Robert. 2003. *DotLRN: Sistema de gestión de la enseñanza*. Consultada en: <http://www.weg.ee.usyd.edu.au/people/rafa/papers/IRICE2003/DotLRN-IRICE3.pdf>.
- [Open] <http://openacs.org/projects/dotLRN/>.
- [CG04] Calvo A. Rafael, Ghiglione Ernie. *The Learning Management System Architecture* Consultada en: <http://www.weg.ee.usyd.edu.au/people/rafa/openacs/dotLRNArch.pdf>.
- [Post] <http://www.postgresql.org/>.
- [Open] <http://openacs.org>.
- [rubic] [http://rubick.com:8002/openacs/getting\\_started/](http://rubick.com:8002/openacs/getting_started/).
- [Open] <http://openacs.org/projects/dotLRN/dotLRN-doc/userdocs>.

- [TR02] Thomas Ruth. 2002. *Interactivity & Simulations in e-Learning*. Consultada en: <http://www.jelsim.org/resources/whitepaper.pdf>.
- [AF01] Anido Luís, Llamas Martín and Fernández J. Manuel. 2001. *Aplicación del Proceso Unificado de Modelo al diseño de Arquitecturas Software para Sistemas de Teleformación dirigidos por Estándares*. Consultada en: Primer Congreso Iberoamericano de telemática. Memorias. Cartagena de Indias, Agosto 2003.