



INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

**“SISTEMA DE ENTRENAMIENTO VIRTUAL
BASADO EN WEB USANDO COMPONENTES
DE SOFTWARE”**

TESIS

**QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS
CON ESPECIALIDAD EN COMPUTACIÓN**

P R E S E N T A

ING. ADRIANA NATALIA RAMÍREZ SALAZAR

**DIRECTORES DE TESIS: M. EN C. RUBÉN PEREDO VALDERRAMA
M. EN C. SERGIO SANDOVAL REYES**

México, D.F.

Noviembre, 2010



ÍNDICE

RESUMEN	1
ABSTRACT	2
AGRADECIMIENTOS	3
CAPÍTULO 1. INTRODUCCIÓN	11
1.1 ANTECEDENTES	11
1.2 PLANTEAMIENTO DEL PROBLEMA	12
1.3 OBJETIVOS	13
1.3.1 OBJETIVO GENERAL	13
1.3.2 OBJETIVOS ESPECÍFICOS	13
1.4 JUSTIFICACIÓN	14
1.5 BENEFICIOS ESPERADOS	16
1.6 TECNOLOGÍAS PROPUESTAS	16
1.7 ALCANCES Y LÍMITES	17
1.8 ORGANIZACIÓN DE LA TESIS	18
CAPÍTULO 2. ESTADO DEL ARTE DE EDUCACIÓN VIRTUAL	19
2.1 OBJETOS DE APRENDIZAJE	19
2.2 ENTRENAMIENTO	20
2.3 SISTEMAS DE ENTRENAMIENTO EN LA WEB	20
2.3.1 ADOBE ACROBAT CONNECT PRO	20
2.3.2 ELECTA LIVE	22
2.3.3 WIZIQ	24
2.3.4 APRENDIZAJE DISTRIBUIDO AVANZADO	25
2.3.4.1 REQUERIMIENTOS DE INSTALACIÓN DE SCORM	26
2.3.4.2 VENTAJAS DE SCORM	28
2.3.4.3 MODELO DE DATOS SCORM 2004	28
2.3.5 SALÓN VIRTUAL BASADO EN WEB	30
2.3.6 MOODLE	32
2.3.7 SISTEMA DE CAMPUS VIRTUAL POLITÉCNICO	35
2.4 COMPONENTES DE SOFTWARE	37
2.5 AGENTES INTELIGENTES	38
2.5.1 CARACTERÍSTICAS DE LOS AGENTES INTELIGENTES	38
2.6 ARQUITECTURA IEEE 1484-LTSA	38
2.7 RESUMEN	39
CAPÍTULO 3. ANÁLISIS Y DISEÑO DE SEVIRTUAL	40
3.1 DIAGRAMA DE CASOS DE USO Y SECUENCIA	40
3.1.1 DIAGRAMA DE CASOS DE USO GENERAL DEL SISTEMA	40
3.1.2 ANÁLISIS DEL CASO DE USO GENERAL DEL SISTEMA	41
3.1.2.1 DIAGRAMA DE SECUENCIA DEL CU: INICIAR SESIÓN	42
3.1.3 DIAGRAMA DE CASOS DE USO DEL ADMINISTRADOR	42
3.1.4 ANÁLISIS DEL CASO DE USO DEL ADMINISTRADOR	43
3.1.4.1 DIAGRAMA DE SECUENCIA DEL CU: ALTA DE USUARIO	44
3.1.4.2 DIAGRAMA DE SECUENCIA DEL CU: BORRAR USUARIO	45
3.1.4.3 DIAGRAMA DE SECUENCIA DEL CU: MODIFICAR USUARIO	46
3.1.5 DIAGRAMA DE CASOS DE USO DEL PROFESOR	47
3.1.6 ANÁLISIS DEL CASO DE USO DEL PROFESOR	47
3.1.6.1 DIAGRAMA DE SECUENCIA DEL CU: CREAR CURSO	48
3.1.6.2 DIAGRAMA DE SECUENCIA DEL CU: AGREGAR NUEVO TEMA	49
3.1.6.3 DIAGRAMA DE SECUENCIA DEL CU: SELECCIONAR COMPONENTE	50
3.1.6.4 DIAGRAMA DE SECUENCIA DEL CU: GUARDAR TEMA	50
3.1.6.5 DIAGRAMA DE SECUENCIA DEL CU: SUBIR MULTIMEDIO	52
3.1.6.6 DIAGRAMA DE SECUENCIA DEL CU: PREVISUALIZAR CURSO	54
3.1.6.7 DIAGRAMA DE SECUENCIA DEL CU: VER CURSOS CREADOS	56

3.1.6.8 DIAGRAMA DE SECUENCIA DEL CU: REGISTRAR CURSO	57
3.1.6.9 DIAGRAMA DE SECUENCIA DEL CU: ELIMINAR CURSO	58
3.1.6.10 DIAGRAMA DE SECUENCIA DEL CU: EDITAR CURSO	59
3.1.6.11 DIAGRAMA DE SECUENCIA DEL CU: MODIFICAR TEMA	59
3.1.6.12 DIAGRAMA DE SECUENCIA DEL CU: MODIFICAR CONTENIDO	60
3.1.7 DIAGRAMA DE CASOS DE USO DEL ALUMNO	63
3.1.8 ANÁLISIS DEL CASO DE USO DE ALUMNO	64
3.1.8.1 DIAGRAMA DE SECUENCIA DEL CU: VER CUSOS REGISTRADOS	64
3.1.8.2 DIAGRAMA DE SECUENCIA DEL CU: INSCRIBIR CURSO	65
3.1.8.3 DIAGRAMA DE SECUENCIA DEL CU: VER CURSOS INSCRITOS	66
3.1.8.4 DIAGRAMA DE SECUENCIA DEL CU: VISUALIZAR CURSO	68
3.1.8.5 DIAGRAMA DE SECUENCIA DEL CU: DAR DE BAJA CURSO	70
3.2 DIAGRAMA DE CLASES	71
3.2.1 DIAGRAMA DE CLASES DEL SERVICIO: ELIMINARCURSOACTION	71
3.2.2 DIAGRAMA DE CLASES DEL SERVICIO: ELIMINARINSCURACTION	71
3.2.3 DIAGRAMA DE CLASES DEL SERVICIO: GUARDARCURSOACTION	72
3.2.4 DIAGRAMA DE CLASES DEL SERVICIO: GUARDARINFOACTION	73
3.2.5 DIAGRAMA DE CLASES DEL SERVICIO: GUARDARTEXTOACTION	73
3.2.6 DIAGRAMA DE CLASES DEL SERVICIO: INSCRIBIRCURSOACTION	74
3.2.7 DIAGRAMA DE CLASES DEL SERVICIO: LOGINACTION	74
3.2.8 DIAGRAMA DE CLASES DEL SERVICIO: OBTENERCURSOSACTION	75
3.2.9 DIAGRAMA DE CLASES DEL SERVICIO: OBTENERCURSOSESTACTION	76
3.2.10 DIAGRAMA DE CLASES DEL SERVICIO: REGISTRARUSUARIOACTION	76
3.2.11 DIAGRAMA DE CLASES DEL SERVICIO: UPLOADACTION	77
3.2.12 DIAGRAMA DE CLASES DEL SERVICIO: USUARIOACTION	77
3.2.13 DIAGRAMA DE CLASES DEL SERVICIO: CURSODAO	78
3.2.14 DIAGRAMA DE CLASES DEL SERVICIO: USUARIODAO	78
3.3 DIAGRAMA DE PAQUETES	79
3.3.1 DIAGRAMA DE PAQUETES DEL MIDDLEWARE DEL SERVIDOR	79
3.3.2 DIAGRAMA DE PAQUETES DE LA APLICACIÓN DEL CLIENTE	80
3.4 DIAGRAMA DE COMPONENTES	81
3.4.1 DIAGRAMA DE COMPONENTES DEL PAQUETE: CURSO	81
3.4.2 DIAGRAMA DE COMPONENTES DEL PAQUETE: PROFESOR	82
3.4.3 DIAGRAMA DE COMPONENTES DEL PAQUETE: ALUMNO	83
3.5 ESTRUCTURA DEL CURSO	83
3.6 RESUMEN	84
CAPÍTULO 4. IMPLEMENTACIÓN DE SEVIRTUAL	85
4.1 PATRÓN DE COMPOSICIÓN	85
4.1.1 VENTAJAS DEL PATRÓN DE COMPOSICIÓN	85
4.2 ARQUITECTURA MODELO VISTA CONTROLADOR	87
4.3 ARQUITECTURA MULTIAGENTES	88
4.4 MÓDULOS DE SEVIRTUAL	89
4.4.1 MÓDULO DE LOGIN	90
4.4.1.1 RIA DEL CLIENTE	90
4.4.1.2 MIDDLEWARE DEL SERVIDOR	91
4.4.2 MÓDULO DEL ADMINISTRADOR	91
4.4.2.1 RIA DEL CLIENTE	91
4.4.2.2 MIDDLEWARE DEL SERVIDOR	92
4.4.3 MÓDULO DEL PROFESOR	92
4.4.3.1 RIA DEL CLIENTE	93
4.4.3.2 MIDDLEWARE DEL SERVIDOR	97
4.4.4 MÓDULO DEL ALUMNO	99
4.4.4.1 RIA DEL CLIENTE	99
4.4.4.2 MIDDLEWARE DEL SERVIDOR	104

4.4.5 MÓDULO DEL MAS	105
4.5 TECNOLOGÍAS DE SOFTWARE	110
4.5.1 APACHE STRUTS 2	110
4.5.2 ADOBE FLEX 3	110
4.5.3 BLAZEDS	111
4.5.4 ULTRAVNC	111
4.6 ESTRUCTURA DEL CURSO	111
4.7 ESTRUCTURA DEL EXÁMEN Y PREGUNTAS	112
4.8 DIAGRAMA ENTIDAD-RELACIÓN	113
4.9 RESUMEN	114
CAPÍTULO 5. PRUEBAS Y RESULTADOS DE SEVIRTUAL	115
5.1 PRUEBAS DE SEVIRTUAL	115
5.2 RESULTADOS DE PRUEBAS DE SEVIRTUAL	137
5.2 COMPARATIVA ENTRE SISTEMAS DE ENTRENAMIENTO	137
5.3 RESUMEN	139
CAPÍTULO 6. CONCLUSIONES	140
6.1 LOGROS ALCANZADOS	140
6.2 APORTACIONES	142
6.3 TRABAJO FUTURO	143
ANEXO A. MANUAL DE OPERACIÓN	144
BIBLIOGRAFÍA	
REFERENCIAS WEB	

ÍNDICE DE TABLAS Y FIGURAS

Tabla 1.1 Usuarios conectados a la Internet en América Latina	15
Tabla 2.1 Elementos del modelo de datos	29
Tabla 3.1 Análisis del caso de uso general del sistema	41
Tabla 3.2 Análisis del caso de uso del administrador	43
Tabla 3.3 Análisis del caso de uso: Alta de usuario	43
Tabla 3.4 Análisis del caso de uso: Borrar usuario	44
Tabla 3.5 Análisis del caso de uso: Modificar usuario	45
Tabla 3.6 Análisis del caso de uso del profesor	47
Tabla 3.7 Análisis del caso de uso: Crear curso	48
Tabla 3.8 Análisis del caso de uso: Agregar nuevo tema	48
Tabla 3.9 Análisis del caso de uso: Seleccionar componente	49
Tabla 3.10 Análisis del caso de uso: Guardar tema	50
Tabla 3.11 Análisis del caso de uso: Subir multimedia	51
Tabla 3.12 Análisis del caso de uso: Previsualizar curso	52
Tabla 3.13 Análisis del caso de uso: Ver cursos creados	56
Tabla 3.14 Análisis del caso de uso: Registrar curso	57
Tabla 3.15 Análisis del caso de uso: Eliminar curso	57
Tabla 3.16 Análisis del caso de uso: Editar curso	58
Tabla 3.17 Análisis del caso de uso: Modificar tema	59
Tabla 3.18 Análisis del caso de uso: Modificar contenido	60
Tabla 3.19 Análisis del caso de uso: Subir video curso	61
Tabla 3.20 Análisis del caso de uso: Crear pregunta	61
Tabla 3.21 Análisis del caso de uso: Crear examen	62
Tabla 3.22 Análisis del caso de uso: Agregar recomendaciones	63
Tabla 3.23 Análisis del caso de uso del alumno	64
Tabla 3.24 Análisis del caso de uso: Ver cursos registrados	64
Tabla 3.25 Análisis del caso de uso: Inscribir curso	65
Tabla 3.26 Análisis del caso de uso: Ver cursos inscritos	66
Tabla 3.27 Análisis del caso de uso: Visualizar curso	66
Tabla 3.28 Análisis del caso de uso: Dar de baja curso	70
Tabla 4.1 Componente Login.mxml	90
Tabla 4.2 Action login.action	91
Tabla 4.3 Componente FormularioRegistro.mxml	91
Tabla 4.4 Action registrarUsuario.action	92
Tabla 4.5 Action buscarUsuario.action	92
Tabla 4.6 Action actualizarUsuario.action	92
Tabla 4.7 Action eliminarUsuario.action	92
Tabla 4.8 Componente NuevoCurso.mxml	93
Tabla 4.9 Componente EditarPregunta.mxml	94
Tabla 4.10 Componente PopUpSubirMult.mxml	95
Tabla 4.11 Componente RecomendacionTema.mxml	95
Tabla 4.12 Componente MisPreguntas.mxml	96
Tabla 4.13 Componente Texto.mxml	96
Tabla 4.14 Componente LlenadoBlanco.mxml	96
Tabla 4.15 Componente VerCursos.mxml	97
Tabla 4.16 Action guardarCurso.action	97
Tabla 4.17 Action guardarInfo.action	98
Tabla 4.18 Action guardarTexto.action	98
Tabla 4.19 Action cursosProfesor.action	98
Tabla 4.20 Action eliminaCurso.action	98
Tabla 4.21 Action completarCurso.action	98
Tabla 4.22 Action enLinea.action	99
Tabla 4.23 Action guardarPregunta.action	99

Tabla 4.24 Action preguntasProf.action	99
Tabla 4.25 Componente MisCursos.mxml	99
Tabla 4.26 Componente InsCurso.mxml	100
Tabla 4.27 Componente VerCurso.mxml	100
Tabla 4.28 Componente AreaContenido.mxml	101
Tabla 4.29 Componente VerMultimedios.mxml	101
Tabla 4.30 Componente BarraNavegacion.mxml	101
Tabla 4.31 Componente Pizarron.mxml	102
Tabla 4.32 Componente Chat.mxml	103
Tabla 4.33 Componente Escritorio.mxml	103
Tabla 4.34 Componente Examen.mxml	104
Tabla 4.35 Action cursosTodos.action	104
Tabla 4.36 Action inscribirCurso.action	104
Tabla 4.37 Action cursosEstudiante.action	105
Tabla 4.38 Action desconectaUsuario.action	105
Tabla 4.39 Action obtenerIP.action	105
Tabla 4.40 Estructura del archivo ADF del sistema: SEVirtualPlan.agent.xml	105
Tabla 4.41 Planes de SEVirtualPlan.java	107
Tabla 4.42 Plan MetricasPlan.java para registro en la base de datos	107
Tabla 4.43 Plan de reconfiguración: ReconfigurarPlan.java	107
Tabla 4.44 Plan de retroalimentación: RetroalimentarPlan.java	108
Tabla 4.45 Estructura de meta datos para la organización del curso	111
Tabla 4.46 Estructura de meta datos para el tipo de componente: título doble texto	112
Tabla 4.47 Estructura de meta datos para la evaluación	112
Tabla 4.48 Estructura de meta datos para el tipo de pregunta: llenado en blanco	113
Tabla 5.1 Comparativo de Sistemas de entrenamiento basados en <i>Web</i>	138
Figura 2.1 "Adobe Acrobat Connect Pro"	21
Figura 2.2 "eLecta Live"	23
Figura 2.3 "WiZiQ"	25
Figura 2.4 "SCORM 2004 3rd Edition Sample RTE"	27
Figura 2.5 Uso del Modelo de Datos y el API	29
Figura 2.6 Modelo del Sistema basado en agentes	31
Figura 2.7 Sistema Virtual Classroom	32
Figura 2.8 Moodle perfil "Administrador"	33
Figura 2.9 Moodle perfil "Profesor" y "Estudiante"	35
Figura 2.10 Sistema de Campus Virtual Politécnico	36
Figura 2.11 Arquitectura IEEE 1484 – LTSA	39
Figura 3.1 Diagrama del caso de uso general del sistema	41
Figura 3.2 Diagrama de secuencia del caso de uso: Iniciar sesión	42
Figura 3.3 Diagrama de casos de uso del administrador	42
Figura 3.4 Diagrama de secuencia del caso de uso: Alta de usuario	44
Figura 3.5 Diagrama de secuencia del caso de uso: Borrar usuario	45
Figura 3.6 Diagrama de secuencia del caso de uso: Modificar usuario	46
Figura 3.7 Diagrama de casos de uso del profesor	47
Figura 3.8 Diagrama de secuencia del caso de uso: Crear curso	48
Figura 3.9 Diagrama de secuencia del caso de uso: Agregar nuevo tema	49
Figura 3.10 Diagrama de secuencia del caso de uso: Seleccionar componente	50
Figura 3.11 Diagrama de secuencia del caso de uso: Guardar tema	51
Figura 3.12 Diagrama de secuencia del caso de uso: Subir multimedia	52
Figura 3.13 Diagrama de secuencia del caso de uso: Previsualizar curso	55
Figura 3.14 Diagrama de secuencia del caso de uso: Ver cursos creados	56
Figura 3.15 Diagrama de secuencia del caso de uso: Registrar curso	57
Figura 3.16 Diagrama de secuencia del caso de uso: Eliminar curso	58
Figura 3.17 Diagrama de secuencia del caso de uso: Editar curso	59
Figura 3.18 Diagrama de secuencia del caso de uso: Modificar tema	60

Figura 3.19 Diagrama de secuencia del caso de uso: Modificar contenido	61
Figura 3.20 Diagrama de casos de uso del alumno	63
Figura 3.21 Diagrama de secuencia del caso de uso: Ver cursos registrados	65
Figura 3.22 Diagrama de secuencia del caso de uso: Inscribir curso	65
Figura 3.23 Diagrama de secuencia del caso de uso: Ver cursos inscritos	66
Figura 3.24 Diagrama de secuencia del caso de uso: Visualizar curso	69
Figura 3.25 Diagrama de secuencia del caso de uso: Dar de baja curso	70
Figura 3.26 Diagrama de clases del servicio: EliminarCursoAction	71
Figura 3.27 Diagrama de clases del servicio: EliminarInsCurAction	72
Figura 3.28 Diagrama de clases del servicio: GuardarCursoAction	72
Figura 3.29 Diagrama de clases del servicio: GuardarInfoAction	73
Figura 3.30 Diagrama de clases del servicio: GuardarTextoAction	74
Figura 3.31 Diagrama de clases del servicio: InscribirCursoAction	74
Figura 3.32 Diagrama de clases del servicio: InscribirCursoAction	75
Figura 3.33 Diagrama de clases del servicio: ObtenerCursosAction	75
Figura 3.34 Diagrama de clases del servicio: ObtenerCursosAction	76
Figura 3.35 Diagrama de clases del servicio: RegistrarUsuarioAction	77
Figura 3.36 Diagrama de clases del servicio: UploadAction	77
Figura 3.37 Diagrama de clases del servicio: UsuarioAction	78
Figura 3.38 Diagrama de clases de la clase: CursoDAO	78
Figura 3.39 Diagrama de clases de la clase: UsuarioDAO	79
Figura 3.40 Diagrama de paquetes del middleware del Servidor	80
Figura 3.41 Diagrama de paquetes de la aplicación del Cliente	81
Figura 3.42 Diagrama de componentes para el paquete: curso	82
Figura 3.43 Diagrama de componentes para el paquete: profesor	82
Figura 3.44 Diagrama de componentes para el paquete: alumno	83
Figura 3.45 Estructura de metadatos para la organización del curso	84
Figura 3.46 Estructura de metadatos para el tipo de componente: título doble texto	84
Figura 4.1 Diagrama de clases del patrón de composición	86
Figura 4.2 Arquitectura MVC y Cliente/Servidor	88
Figura 4.3 Tutor virtual con el MAS	89
Figura 4.4 Plataforma JADEX	109
Figura 4.5 Diagrama relacional de SEVirtual	114
Figura 5.1 Información del nuevo usuario	116
Figura 5.2 Usuario registrado	116
Figura 5.3 Login – profesor	117
Figura 5.4 Nuevo curso – profesor	117
Figura 5.5 Tema: “Introducción” – profesor	118
Figura 5.6 Tema “Introducción” guardado – profesor	118
Figura 5.7 Login - primer alumno	119
Figura 5.8 Curso Photoshop - primer alumno	119
Figura 5.9 Imagen visualizada - primer alumno	120
Figura 5.10 Login - segundo alumno	121
Figura 5.11 Curso Photoshop - segundo alumno	121
Figura 5.12 Reproducción de audio mp3 - segundo alumno	122
Figura 5.13 Login - tercer alumno	123
Figura 5.14 Curso: Photoshop - tercer alumno	123
Figura 5.15 Reproducción de video flv - tercer alumno	124
Figura 5.16 Búsqueda del usuario – administrador	124
Figura 5.17 Actualización del usuario – administrador	125
Figura 5.18 Búsqueda del estudiante modificado – administrador	125
Figura 5.19 Eliminación del estudiante – administrador	126
Figura 5.20 Multimedia agregado – profesor	126
Figura 5.21 Video agregado – profesor	127
Figura 5.22 Explicación del video agregada – profesor	127
Figura 5.23 Sincronización del video con los temas – profesor	128

Figura 5.24 Descarga del recurso – primer alumno	129
Figura 5.25 Uso del chat – primer alumno	129
Figura 5.26 Uso del chat – segundo alumno	130
Figura 5.27 Uso del pizarrón – segundo alumno	131
Figura 5.28 Compartición de escritorio – primer alumno	131
Figura 5.29 Visualización de escritorio – segundo alumno	132
Figura 5.30 Evaluación – tercer alumno	132
Figura 5.31 Retroalimentación – tercer alumno	133
Figura 5.32 Pregunta incorrecta – tercer alumno	134
Figura 5.33 Cursos reconfigurados – tercer alumno	134
Figura 5.34 Curso: “Photoshop” reconfigurado– tercer alumno	135
Figura 5.35 Curso: Pregunta: falso-verdadero – profesor	136
Figura 5.36 Pregunta guardada – profesor	136
Figura 5.37 Examen guardado – profesor	137
Figura A.1 Login	144
Figura A.2 Interfaz del administrador	145
Figura A.3. Registrar y buscar usuarios	146
Figura A.4. Actualizar y eliminar usuarios	147
Figura A.5. Menú del profesor	148
Figura A.6. Eliminar curso	148
Figura A.7. Editar curso	149
Figura A.8. Nuevo curso	150
Figura A.9. Editar tema	150
Figura A.10. Guardar Tema	151
Figura A.11. Agregar multimedios	152
Figura A.12. Subir video explicativo del curso	153
Figura A.13. Agregar pregunta	154
Figura A.14. Crear examen	157
Figura A.15. Previsualizar examen	157
Figura A.16. Guardar examen	158
Figura A.17. Recomendar	158
Figura A.18. Conectar al curso	159
Figura A.19. Previsualizar curso	160
Figura A.20. Registrar el curso	160
Figura A.21. Terminar el curso	161
Figura A.22. Menú del alumno	162
Figura A.23. Eliminar curso inscrito	162
Figura A.24. Inscribir curso disponible	163
Figura A.25. Visualizar curso: Photoshop	163
Figura A.26. Componente: índice	164
Figura A.27. Componente: barra de navegación	164
Figura A.28. Navegación a través del árbol de temas	165
Figura A.29. Navegación a través de la barra de navegación	165
Figura A.30. Navegación a través de la reproducción del video	166
Figura A.31. Componente: multimedios	167
Figura A.32. Componente: comunicación	168
Figura A.33. Componentes: pizarrón y chat	168
Figura A.34. Componentes: compartir escritorio y descargar recursos	169
Figura A.35. Componente: evaluación	171
Figura A.36. Siguiendo pregunta	171
Figura A.37. Resultado de la evaluación	172
Figura A.38. Pregunta: “Photoshop edita.”	173
Figura A.39. Cursos reconfigurados	173
Figura A.40. Curso reconfigurado: “Photoshop”	174

RESUMEN

SEVirtual (Sistema de Entrenamiento Virtual basado en *Web* usando componentes de software) es un sistema que automatiza la generación de cursos de entrenamiento en línea bajo los paradigmas de Educación Basada en *Web* y Programación Orientada a Componentes, siendo una herramienta de apoyo para el profesor reduciendo la complejidad en cuanto a aspectos técnicos en la generación de cursos de entrenamiento.

SEVirtual también sirve de apoyo para el estudiante ya que al simular un profesor virtual implementado a través de un sistema multi-agente permite dar retroalimentación dinámica, recomendaciones al alumno y la reconfiguración del curso en base a los resultados obtenidos durante su evaluación.

Para hacerlo más versátil se integraron dentro del sistema diversos componentes interactivos como el de: chat, pizarrón, evaluación, compartición de escritorio y recursos, generación de contenido y visualización del curso, haciendo uso de meta-datos para la configuración de los componentes en tiempo de ejecución y la separación del contenido de la parte de navegación de los cursos, todo esto utilizando herramientas de software libre para la reducción de costos de licenciamiento.

ABSTRACT

SEVirtual automates the creation of online training courses under the paradigms of Web-Based Education and Component-Oriented Programming, being a support tool for teachers reducing the complexity of technical aspects in the generation of online training courses.

SEVirtual also supports the student, simulating a virtual tutor implemented through a multi-agent system whereby is able to provide dynamic feedback, recommendations to the student and the reconfiguration of his course based on the results obtained during the assessment.

Diverse interactive components were integrated into the system such as: chat, whiteboard, assessment, desktop sharing and resources, content generation, etc, using meta-data for the configuration of the runtime components and the separation of content from the navigation of the courses, all this using open software to reduce licensing costs.

CAPÍTULO 1

INTRODUCCIÓN

En el presente capítulo se hablan de algunos antecedentes importantes del trabajo de tesis, entre ellos la Educación Basada en *Web*, se plantea el problema a resolver, así como el objetivo general y los específicos.

Además se detalla la justificación del por qué de este trabajo, algunas aportaciones principales que se obtuvieron con el desarrollo de este proyecto, así como las tecnologías que se utilizaron durante su elaboración y los alcances y límites del sistema.

1.1 Antecedentes

Debido a la existencia de tecnologías innovadoras como Internet, que ha causado un gran impacto en la sociedad, la educación ha cambiado considerablemente; ya que por ejemplo gran parte de la información que se consulta ya se encuentra en un formato electrónico y está al alcance de cualquier persona que cuente con una conexión a la *Web*.

Gran parte de los estudiantes en busca de la educación, prefieren estudiar a su propio ritmo, en cualquier momento y en cualquier lugar, estos puntos importantes son tres de las principales ventajas del paradigma de **Educación Basada en Web** (*Web Based Education*, **WBE** por sus siglas en inglés), como la educación se distribuye a través de la *Web*, está disponible en cualquier momento y en cualquier lugar, esto permite al estudiante avanzar a su propio ritmo, además de que el estudiante puede repasar aquellos temas que se le dificulten más las veces que el desee. En la educación tradicional esto no es posible ya que el profesor lleva el ritmo de todo el grupo y no se puede estar deteniendo en cada tema.

Los factores sociales, económicos y culturales han propiciado este cambio en la educación, como por ejemplo el hecho de que muchos estudiantes se ven obligados a abandonar sus estudios debido a problemas económicos o el hecho de la sobredemanda a las instituciones públicas por parte de los estudiantes en busca de un lugar donde poder realizar sus estudios, las cuales no se dan abasto por la falta de infraestructura. Debido a ello se busca crear soluciones que apoyen a la educación actual haciendo uso del paradigma **WBE**.

Las líneas de investigación actuales en cuanto a lo que es la **WBE** se centran en el reuso y comercialización de los objetos de aprendizaje, que son recursos digitales que apoyan el aprendizaje, la secuenciación dinámica de los materiales de aprendizaje, así como la separación del contenido de la parte de control o navegación, las cuales será implementadas en el trabajo de tesis.

Existen tres principales iniciativas internacionales las cuales son líderes en la educación en línea: la Iniciativa **del Aprendizaje Abierto del MIT** (*Open Knowledge Initiative*, **OKI** por sus siglas en inglés), el **Aprendizaje Distribuido Avanzado** (*Advanced Distributed Learning*, **ADL** por sus siglas en inglés) y el **Consortio de Aprendizaje Global IMS** (*Global Learning Consortium*, **IMS** por sus siglas en inglés), el presente trabajo de tesis se basa en la iniciativa de **ADL** ya que se considera el estándar más avanzado en cuanto a educación en línea.

El alcance del trabajo de tesis abarca desde una herramienta de ayuda para el profesor, la cual le permita automatizar la generación de cursos de entrenamiento de cualquier tema sin que tenga que preocuparse por razones técnicas, simplemente teniendo sus materiales a la mano, así como una herramienta inteligente para el alumno.

1.2 Planteamiento del problema

El problema a resolver es el siguiente:

- Diseñar e implementar un Sistema de Entrenamiento Virtual Basado en Web usando Componentes de Software.

Se propone diseñar e implementar una herramienta de apoyo para el profesor y el estudiante, utilizando componentes de software y patrones de diseño, la cual le permita automatizar la generación de cursos de entrenamiento acerca de cualquier tema, sin que el profesor tenga que preocuparse por aspectos técnicos simplemente teniendo a la mano sus materiales, el sistema le permitirá generar el curso, donde el profesor podrá proponer su estructura, subir multimedia, un video/audio y texto explicativo, así como el contenido del mismo.

A través del sistema el estudiante tendrá acceso a los cursos que él desee y sean de su interés, contará con un sistema asistencial inteligente, el cual en la parte de su evaluación le proporcionará una retroalimentación dinámica, una reconfiguración dinámica del curso con los temas que se le hayan dificultado, así como una recomendación de ejercicios, libros o páginas de Internet de dichos temas.

El presente trabajo de tesis se basa en el **Modelo de Referencia de Objetos de Contenido Compartido** (*Sharable Content Object Reference Model*, **SCORM** por sus siglas en inglés) de **ADL**, en el cual se pretende proponer una arquitectura de meta datos para la organización de la estructura de los cursos de entrenamiento, permitiendo la reutilización de los componentes de software, así como separar el contenido de la parte de la navegación del curso, haciendo uso de las ventajas de los componentes de software y de los patrones de diseño, particularmente el patrón de composición.

La parte inteligente del sistema se pretende llevar a cabo a través de un sistema multi-agente, los cuales a través de ciertas métricas (modelo de datos) puedan percibir su ambiente y actuar en función de sus objetivos.

1.3 Objetivos

A continuación se presentan los objetivos principales del trabajo de tesis:

1.3.1 Objetivo General

Desarrollar un Sistema de Entrenamiento Virtual de **EBW** basado en el patrón de composición y la arquitectura Modelo-Vista-Controlador (Model-View-Controller, **MVC** por sus siglas en inglés), para generar componentes de software ricos en multimedia, separando el contenido de la navegación, dentro de un ambiente asistencial basado en agentes.

1.3.2 Objetivos Específicos

El objetivo general se divide en los siguientes objetivos específicos:

- Desarrollo del módulo de inicio de sesión.
- Desarrollo del módulo del administrador (altas, bajas y modificaciones).
- Desarrollo del módulo del profesor.
- Desarrollo del módulo del estudiante.
- Desarrollo del módulo de separación de contenido y navegación.
- Desarrollo de una arquitectura de meta datos.
- Desarrollo del módulo del sistema de componentes que permita generar materiales interactivos.
 - Generación de contenido.
 - Audio/Video sincronizado con contenido.
 - Evaluación.
 - Chat.
 - Pizarrón.
 - Compartición de escritorio y recursos.
- Integración de unidades de composición.
- Desarrollo del módulo del sistema de agentes.

1.4 Justificación

El artículo “**Electronics and the dim future of university**” [46], dice: “Si bien las nuevas tecnologías de comunicación pueden reforzar la investigación, también pueden debilitar a las grandes instituciones de aprendizaje, las universidades”; si bien es cierto que la tecnología ha venido a apoyar a las instituciones de aprendizaje más que a debilitarlas, es por ello que dichas instituciones deben de implementar la tecnología no solo para aprovechar sus ventajas sino también para ser competitivas con otras universidades del país como por ejemplo, el Tecnológico de Monterrey y la Universidad Nacional Autónoma de México.

Las universidades virtuales se han creado con el fin de satisfacer gran parte de la demanda por parte de los estudiantes que no obtienen algún lugar en las instituciones públicas, así como satisfacer la demanda de una educación que se adapte a las necesidades de los estudiantes. Debido a ello instituciones como la UNAM, el Tecnológico de Monterrey y el Instituto Politécnico Nacional han creado sus propias universidades virtuales las cuales en cierto orden son de gran importancia para el país.

A continuación se listan las universidades virtuales más importantes del país:

- Universidad en Línea de la Universidad Nacional Autónoma de México (UNAM)
- Universidad Virtual del Tecnológico de Monterrey (ITESM)
- Campus Virtual del Instituto Politécnico Nacional (IPN)

Se enlistan en dicho orden, ya que debido a un estudio realizado por el Laboratorio de Cibermetría del Consejo Superior de Investigaciones Científicas de España (CSIC) [47], la UNAM se encuentra en el lugar 44 del mundo de las universidades en la Red, el ITESM en el lugar 391, mientras que el IPN en el lugar 943, es por ello que es necesario impulsar la educación en línea en el IPN.

Actualmente el país cuenta con un gran número de usuarios conectados a Internet, como se muestra en la siguiente tabla 1.1, donde el número de usuarios conectados ha ido en aumento, es decir, gran parte de la población ya cuenta con acceso a Internet. No obstante, los sistemas **WBE** pueden ser distribuidos a través de *Intranets* o medios de almacenamiento como los CD.

Tabla 1.1 Usuarios conectados a Internet en América Latina [48] [49]

Usuarios de internet en países seleccionados de América Latina (en millones)			
Región	Junio 1997	Junio 2000	2005 ***
Argentina	0,17	0,90	7,00
Bolivia	0,008	0,035	N/D
Brasil	0,47	10,50*	29,10
Chile	0,20	0,625	2,70
Colombia	0,12	0,60	4,40
Ecuador	0,005	0,02	N/D
México	0,37	2,45	12,70
Paraguay	0,001	0,02	N/D
Perú	0,065	0,40	2,40
Puerto Rico	N/D	0,21 **	N/D
Uruguay	0,009	0,25	N/D
Venezuela			
Otros Latinoamérica	0,00	0,00	
TOTAL	1,453	16,41	66,60

Como parte de la justificación de este trabajo de tesis, se encuentran los siguientes puntos importantes:

- Integrar a personas que, por motivos culturales, sociales o económicos no se adaptan o no tienen acceso a los sistemas convencionales de educación.
- Involucrar en las clases la participación de expertos de otros países.
- Reducir costos.
- Hacer uso de las tecnologías actuales.
- Crear herramientas que automatizan la producción de cursos de entrenamiento.
- Crear sistemas que manejan Inteligencia Artificial.
- Crear sistemas basados en componentes de software.

Debido a falta de herramientas que permitan automatizar la generación de cursos de entrenamiento, se propone el desarrollo de **SEVirtual**, ya que es una herramienta que hereda las ventajas de la programación basada en componentes (las cuales se explican a continuación), los costos de licenciamiento son bajos, implementa el uso de estándares internacionales y algo muy importante, cuenta con un sistema asistencial basado en agentes, el cual permitirá una retroalimentación más inteligente para la parte de evaluación del alumno, recomendaciones de ejercicios, libros y ligas de Internet sobre los temas que se le hayan dificultado más al alumno así como la reconfiguración del curso en base también a dichos temas.

1.5 Beneficios esperados

Además de hacer uso de las ventajas que brinda el paradigma de **EBW** y el patrón de composición, el trabajo de tesis plantea los siguientes beneficios esperados:

- Proponer una arquitectura para un modelo de Agentes.
- Proponer una arquitectura basada en componentes.
- Proponer una arquitectura de meta datos.
- Separar el contenido del curso de la parte de navegación.
- Sincronización entre el contenido y el audio/video explicativo del curso.
- Reducción significativa de costos de licenciamiento.
- Compartir el contenido y el escritorio del usuario.
- Desarrollo de componentes innovadores como: el pizarrón, el chat, etc.
- Construir una herramienta que permita automatizar la generación de cursos en línea, permitiendo la generación de contenidos.
- Construir una herramienta que permita la generación de materiales interactivos.

1.6 Tecnologías propuestas

Las tecnologías que se proponen para el desarrollo del trabajo de tesis, son las siguientes:

- Programación Orientada a Objetos (Java, JSP, Servlets, JB, Struts).
- Programación Orientada a Componentes.
- Multimedia.
- Servidores Web.
- XML.
- Redes.
- Bases de Datos.
- Hibernate, Xalan, Xerces, Action Script, Adobe Flex.

1.7 Alcances y límites

- El sistema necesita de un servidor de escritorio remoto llamado UltraVNC el cual debe de estar ejecutándose en la máquina del participante que desea compartir su escritorio remoto.
- El profesor solo puede generar un examen por curso de entrenamiento.
- Los multimedia que se pueden subir y visualizar son de los siguientes tipos: JPG, GIF, PNG, FLV y MP3.
- El profesor debe dar clic en la opción "Guardar Tema" para persistir los cambios que haya hecho al contenido de algún tema o a la estructura del curso.
- El profesor debe tener sus materiales: contenido, video y texto explicativo, multimedia, información de las preguntas, etc., preparados para crear el curso.
- El profesor no recibe por el momento información como las calificaciones del alumno, estadísticas, reportes, etc., se deja como trabajo futuro.
- En el curso reconfigurado, el alumno no puede visualizar el video ni el texto explicativo del curso.
- Los archivos de los cursos eliminados no son borrados del servidor.
- El alumno al visualizar el curso debe de estar en el área de comunicación para recibir mensajes del chat, imágenes del pizarrón, etc.
- Es necesario instanciar la plataforma JADEX para poner en funcionamiento al profesor virtual.
- Al profesor virtual solo se le envía métricas relacionadas con las preguntas que se ha equivocado y los temas relacionados con estas preguntas.
- El profesor virtual solo se encarga de: reconfigurar el curso en base a los temas que se le han dificultado al alumno, dar retroalimentación dinámica y recomendaciones de páginas de Internet, libros y ejercicios de dichos temas.
- Es necesario configurar los archivos temporales de Internet dentro de Internet Explorer, a través del menú "Herramientas-> Opciones de Internet", en la pestaña "General", dar clic en el botón "Configuración" y por último seleccionar la opción: "Cada vez que visite la página Web".

1.8 Organización de la tesis

El desarrollo del trabajo de tesis se encuentra dividido en 6 capítulos. En el presente capítulo se presentó una introducción al trabajo de tesis, dentro de esta el planteamiento del problema a resolver, los objetivos principales y la justificación del trabajo de tesis. En el capítulo 2 se presenta el marco teórico con las definiciones y aspectos considerados importantes para el desarrollo de la tesis, así como el estado del arte, mostrando una tabla comparativa entre los sistemas de entrenamiento que ya se encuentran en la *Web*, así como sus características.

En el capítulo 3 se muestra el análisis y diseño del sistema, en el cual se presenta la arquitectura general de la aplicación, es decir, los casos de uso principales, así como los diagramas de secuencia y clases principales del sistema, el diagrama de paquetes y componentes, la estructura de los meta datos utilizados y para el diseño de la base de datos se muestra el diagrama Entidad-Relación.

En el capítulo 4 se observa la implementación de todos los módulos del sistema, así como su explicación.

En el capítulo 5 se exponen las pruebas que se le han realizado al sistema de todas sus funcionalidades para cada actor que existe: el profesor, el alumno y el administrador, así como una comparativa del sistema con otros sistemas similares.

En el capítulo 6 se presentan las conclusiones del sistema, donde se detallan los logros que se alcanzaron al desarrollar este trabajo de tesis, las aportaciones y el trabajo futuro.

Por último se integran las referencias bibliográficas recopiladas durante la elaboración de la tesis, y un breve anexo de la operación del sistema.

CAPÍTULO 2

ESTADO DEL ARTE DE EDUCACIÓN VIRTUAL

En el presente capítulo se tratan conceptos importantes relacionados con la educación virtual [9-18], el aprendizaje electrónico [1-4], los sistemas tutores inteligentes [7-8], así como la Educación Basada en Web [5-6, 19-21], a través de la cual se basa el trabajo de tesis, así como los componentes de software que heredan ventajas al sistema, todo esto se lleva a cabo a través de la arquitectura Cliente/Servidor [22] para la comunicación vía Web; también se presenta dentro del marco teórico los principales sistemas de entrenamiento que se encuentran en la Web y una comparación del sistema que se desarrolla como parte de esta investigación.

2.1 OBJETOS DE APRENDIZAJE

Las actuales iniciativas de especificación y estandarización del **eLearning**, como el **SCORM** y los **IMS** (originalmente **IMS** significaba *Instructional Management Systems*, pero ahora oficialmente las siglas carecen de significado), se han preocupado por hacer de los **Objetos de Aprendizaje** (*Learning Objects*, **LO** por sus siglas en inglés), recursos que puedan ser interoperables, es decir, que funcionen y sean comprendidos por distintas plataformas.

Algunas definiciones de los **LO** se presentan a continuación:

- "Son principalmente un conjunto de materiales digitales los cuales como unidad o agrupación permiten o facilitan alcanzar un objetivo educacional" [23].
- "Son cualquier recurso digital que puede ser reutilizado para facilitar el aprendizaje" [23].
- "Son la mínima estructura independiente que contiene un objetivo, una actividad de aprendizaje y un mecanismo de evaluación" [24].
- "Cualquier recurso digital que se puede utilizar como apoyo para el aprendizaje" [23].

Los **Objetos de Aprendizaje**, son recursos digitales que apoyan la educación y que pueden reutilizarse constantemente. Si se consideran las definiciones anteriores, además del objetivo de **SCORM** se tiene, entonces, una gama enorme de recursos digitales, con diferentes niveles de uso que pueden ir, desde apoyar la clase presencial con una imagen, hasta asistir a un curso en línea.

La práctica y las distintas definiciones existentes actualmente ligan el concepto de **Objetos de Aprendizaje** con la idea de "contenidos" o "recursos digitales".

Algunos ejemplos de recursos digitales reutilizables más pequeños incluyen a imágenes o fotos, datos en vivo, cortos de video o audio pregrabados o en vivo, pequeñas porciones de texto, animaciones y pequeñas aplicaciones Web. Ejemplos de recursos digitales reutilizables de mayor tamaño son páginas Web completas que combinen texto, imágenes y otros medios de comunicación [23].

2.2 ENTRENAMIENTO

El **entrenamiento** es un proceso concreto para transmitir información acerca de un tema específico; para entender mejor este concepto a continuación se presentan algunas definiciones de **entrenamiento**:

- “*Es un proceso educativo a corto plazo mediante el cual las personas aprenden conocimientos, aptitudes y habilidades en función de objetivos definidos*” [25].
- “*En términos de una organización, es la educación profesional que adapta al hombre para un cargo o función dentro de una organización, e implica la transmisión de conocimientos, ya sea información de la empresa, de sus productos, servicios, etc.*” [25].
- “*De manera concreta es el proceso de enseñar, comunicar ideas, información y conocimiento. Dar a conocer el estado de algo, formalizar un proceso*” [26].

En segunda medida, el **entrenamiento** implica un desarrollo de habilidades, entendido como un entrenamiento orientado a las tareas y operaciones que van a ejecutarse [25].

2.3 SISTEMAS DE ENTRENAMIENTO EN LA WEB

Algunos Sistemas de Entrenamiento que se encuentran actualmente disponibles en la *Web* son los llamados **Salones Virtuales** (*Virtual Classroom*, **VC** por sus siglas en inglés), los cuales han sido desarrollados por importantes empresas como es el caso de “**Adobe Acrobat Connect Pro**” desarrollado por **Adobe** [27].

Un **Salón Virtual**, es una solución completa de **eLearning** para educadores en línea, la cual puede facilitar la interacción y mejorar la colaboración, el **Salón Virtual** simula el salón de clases real y permite al profesor enseñar a sus alumnos como lo hace en los salones de clases tradicionales [28, 29].

2.3.1 Adobe Acrobat Connect Pro

Adobe presenta una solución a este tipo de sistemas llamada “**Adobe Acrobat Connect Pro**”, el cual es un sistema de alto impacto para conferencias *Web/eLearning*, es dinámico, fácil de usar y está siempre disponible [27].

Es un software que ha ganado distinción como líder en “Gartner’s 2008 Magic Quadrant for Web Conferencing” [30]. Actualmente se encuentra en su séptima edición y ofrece tres ventajas sobre los productos de la competencia:

- **Acceso Instantáneo** — No se requieren descargas para unirse a las reuniones.
- **Alto Impacto** — Una rica interfaz captura la atención de los participantes y proporciona una manera intuitiva para interactuar.

- **En vivo y bajo demanda** — Acrobat Connect Pro es una solución completa para presentaciones en vivo y bajo demanda.

La clara desventaja de este software, es que sus precios son excesivamente altos, los planes que se ofrecen están restringidos al número de usuarios que se conecten, al tiempo de uso y al tipo de características que ofrece.

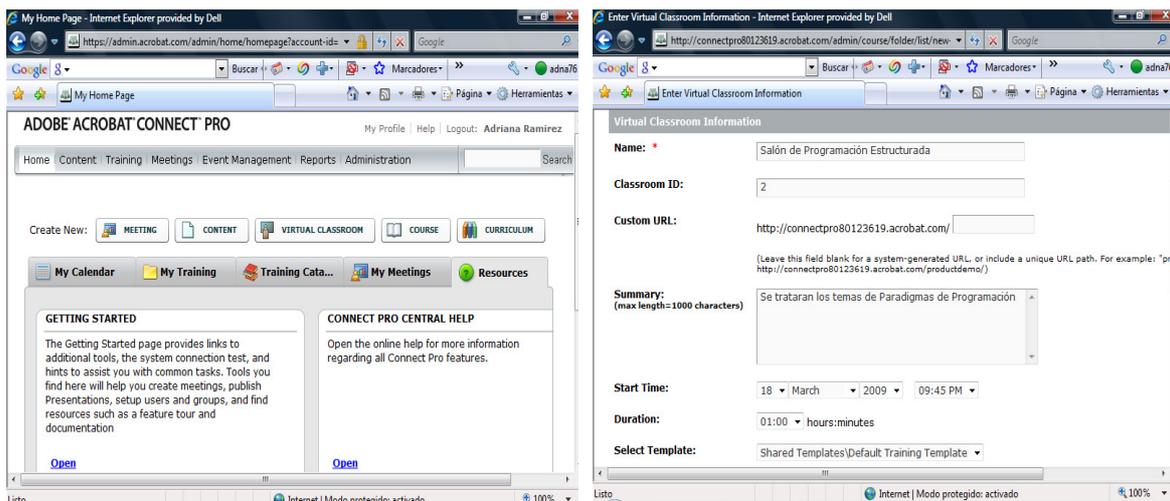
Por ejemplo hay planes de suscripción anual, mensual y pago por uso, es decir Adobe cobra únicamente el tiempo de uso del software.

Hay paquetes para 5 usuarios conectados a la vez con un costo de \$375 US al mes, paquetes para 10 usuarios a la vez con un costo de \$750 US al mes y el Plan pago por uso, el cual tiene un costo de \$0.32 US cada minuto por usuario. Lo cual es una desventaja para los usuarios e Instituciones que desean utilizar este tipo de sistemas ya que se tienen que estar pagando altos costos de licenciamiento.

Algunas de las características que ofrece este sistema son:

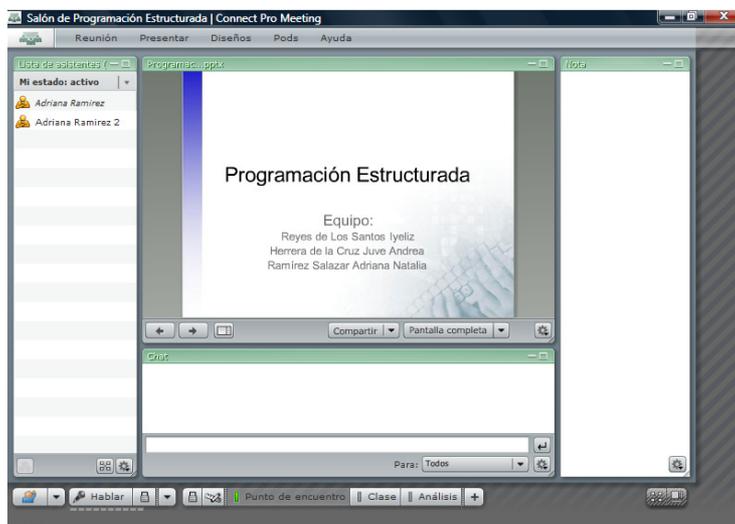
- Voz sobre IP (VoIP), audio y transmisión de video.
- Teleconferencia.
- Mantener reuniones en una organización o utilizarlo como **VC**.
- Compartir contenidos.
- Presentaciones en PowerPoint.
- Compartir escritorio y ventanas.

La **Figura 2.1 (a)** muestra la página principal, a través de la cual es posible crear nuevas reuniones, contenido, salones virtuales, cursos y curriculums. A través de los menús es posible interactuar con las opciones que presenta **Adobe Acrobat Connect Pro**, entre ellas el calendario que muestra los cursos o reuniones agendados, los eventos y reuniones disponibles, así como los recursos como son tutoriales y documentación de **Adobe Acrobat Connect Pro**.



a) Página principal

b) Crear nuevo Salón Virtual



c) Salón Virtual

Figura 2.1. "Adobe Acrobat Connect Pro"

La **Figura 2.1 (b)** muestra la página donde se crean los salones virtuales, en la cual es posible agregar información relacionada al salón virtual que se va a crear, como es el nombre, una pequeña descripción, la fecha y hora de inicio, etc.

Una vez creada una reunión o un salón virtual, es posible acceder a él. La **Figura 2.1 (c)** muestra un **VC** creado, del lado izquierdo de la pantalla se muestra la lista de asistentes, en la parte central se muestra el contenido del curso, en este caso una presentación, en la parte inferior se muestra el chat y del lado derecho un área libre para escribir notas sobre el curso.

2.3.2 eLecta Live

eLecta Live es un Software de **VC** y una solución para conferencias en *Web* tanto para clases en vivo y reuniones en línea. Es una solución poderosa y escalable para Escuelas en línea, Universidades, Corporaciones de entrenamiento y también para maestros. **eLecta Live** puede ser usado tanto de manera independiente o integrada con un Sistema Administrador de Aprendizaje (*Learning Management System*, **LMS** por sus siglas en inglés) y un Sistema Administrador de Estudiantes (*Student Management System*, **SMS** por sus siglas en inglés) [28].

eLecta Live está disponible en diferentes ediciones y configuraciones diseñadas para enseñanza y entrenamiento en línea así como conferencia en *Web* y colaboración en *Web*.

Las configuraciones que ofrece **eLecta Live** son las siguientes:

- Reuniones generales a través de la *Web* rentables - **eLecta Live Easy Meetings**
- Seminarios y Colaboración en línea - **eLecta Live Web Events**
- Enseñanza o asistencia a las personas a través de Internet - **eLecta Live Tutor Edition**
- Soluciones universales para enseñanza, entrenamiento y reuniones en línea - **eLecta Live Web Events**
- Sesiones simultáneas en línea, lecturas en vivo, múltiples equipos - **eLecta Live Professional**

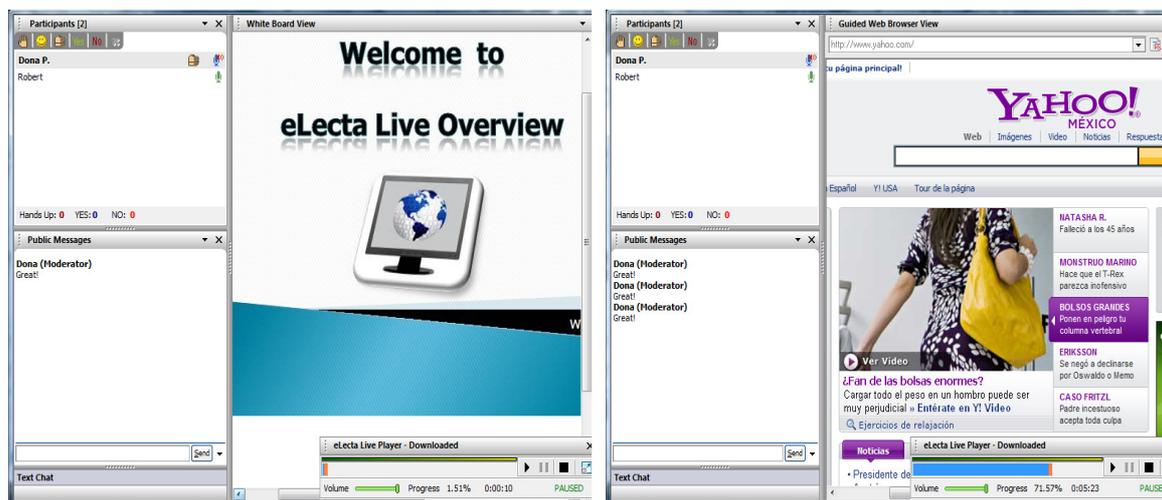
— Enseñanza a través de Internet con varios profesores a la vez - **eLecta Live Professional**

El costo de licenciamiento de **eLecta Live** es de \$25 US por mes, para profesores individuales. Las universidades, escuelas, compañías y comunidades en línea que se interesen en usarlo deben de ponerse en contacto con **eLecta Live**, claro está que el costo sería aún mayor.

Las características que ofrece **eLecta Live** son las siguientes:

- Pizarrones multifuncionales interactivos
- Audio de conferencia integrado
- Full Duplex Voz sobre IP de excelente calidad
- Nada de cargos extras o llamadas telefónicas necesarias
- Mensajes instantáneos y encuestas en línea
- Archivos e imágenes a compartir en tiempo real
- Grabación de sesiones y reproducción
- Presentaciones en PowerPoint a compartir
- Herramientas de marcado y punteros
- Compartición de escritorio y aplicación
- Herramientas para comentarios y de señales
- Video de alta calidad
- Acceso y Control basado en *Web*
- Sistema Avanzado de programación de eventos

La **Figura 2.2 (a)** muestra la página principal de **eLecta Live**, del lado izquierdo se muestra la lista de participantes así como el chat y del lado derecho se muestra el pizarrón con el contenido del curso.



a) Página principal "eLecta Live"

b) Compartir sitios *Web* durante la sesión

Figura 2.2. "eLecta Live"

La **Figura 2.2 (b)** muestra como se comparte un sitio *Web* de manera síncrona durante la sesión, **eLecta Live** también permite compartir el escritorio con los demás usuarios, esta es una gran

característica de los **VC** con la cual es posible visualizar el escritorio de otros usuarios con fines didácticos.

2.3.3 WiZiQ

WiZiQ es un **VC** para aprender o enseñar en vivo. Interactivo sin costos de tiempo y traslado lo que lo hace una alternativa de enseñanza - aprendizaje. Está completamente basado en *Web*, sin necesidad de descarga del software y trabaja en cualquier sistema operativo [31].

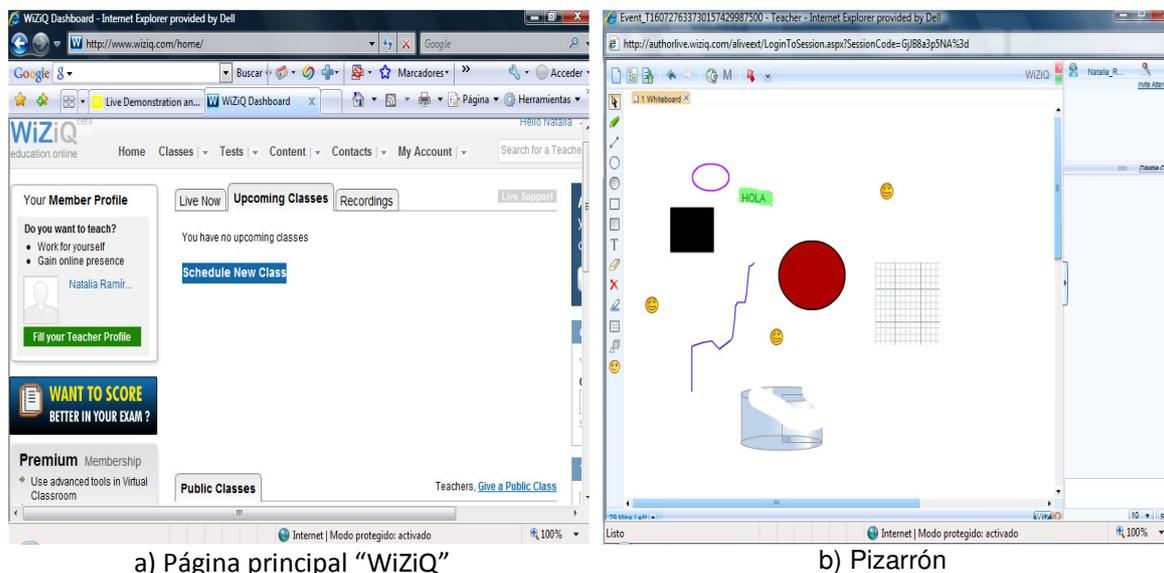
El ambiente *Web* de conferencia colaborativo permite la comunicación de manera síncrona usando audio y video o a través del chat, compartir presentaciones, documentos e imágenes en un pizarrón interactivo.

WiZiQ tiene características *premium* que tienen un cierto costo adicional.

Las características que ofrece **WiZiQ** son las siguientes:

- Selección de solo audio o transmisión de audio y video.
- Permite compartir el Escritorio y la pantalla.
- Todas las clases se graban automáticamente.
- Barra de herramientas completa con funciones matemáticas.
- Permite subir archivos pdf, doc, xls, ppt y flash fácilmente.
- Chat privado (Característica Premium).
- No se requieren descargas para ejecutar el **VC**.
- Comunicación en vivo: chat, conferencia a través de audio y video.
- Permite compartir videos de YouTube de manera sincronizada y cambiar el tamaño de los videos.
- Pizarrones interactivos con múltiples pestañas.
- Permite cortar, cambiar de tamaño, y borrar objetos en el pizarrón.
- Permite marcar el **VC** con el nombre del usuario o el nombre de la organización (Característica Premium).
- Transferencia de los controles de audio, video y pizarrón a los participantes.

WiZiQ permite crear y programar sesiones, invitar contactos, etc. La **Figura 2.3 (a)** muestra su página principal, en la cual mediante los menús es posible acceder a los cursos, al contenido agregado, a los contactos, las clases, etc. Si se ha agendado una nueva clase ésta se muestra en la pestaña de "Próximas Clases" (***Upcoming Classes***).



a) Página principal "WIZIQ"

b) Pizarrón

Figura 2.3. "WIZIQ"

Además **WIZIQ**, permite compartir contenido y archivos. La **Figura 2.3 (b)** muestra el pizarrón el cual es posible utilizar durante la sesión, el pizarrón permite explicar de forma gráfica dudas o comentarios acerca del curso de una manera fácil.

2.3.4 Aprendizaje Distribuido Avanzado

En 1997, la iniciativa **ADL**, fue creada como un programa del Departamento de Defensa de los Estados Unidos y de la Oficina de Ciencia y Tecnología de la Casa Blanca, para desarrollar principios y guías de trabajo necesarias para el desarrollo y la implementación eficiente, efectiva y en gran escala, de formación educativa sobre las nuevas tecnologías *Web*.

El **ADL** emplea un esfuerzo estructurado, adaptativo y colaborativo entre los sectores públicos y privados para desarrollar estándares, herramientas y contenidos de aprendizaje para el entorno de aprendizaje del futuro.

La visión de la iniciativa de **ADL** es proporcionar acceso a aprendizaje de la más alta calidad, que puede adaptarse a las necesidades individuales de las personas y ser entregado de manera rentable, en cualquier momento y en cualquier lugar [32].

El **SCORM** utilizado por **ADL**, propone un entorno de ejecución, un modelo de meta datos y un modelo de la estructura de los cursos [32].

El **SCORM** define un software que describe el modelo de agregación de contenidos, las interrelaciones establecidas entre los componentes de los cursos, los modelos de datos y los protocolos de comunicación, de manera que los objetos definidos en un **LMS** puedan compartirse entre diferentes **LMS**.

Un **LMS** permite la publicación de contenidos, el acceso a los contenidos, la gestión de los recursos y la comunicación entre todos los actores implicados (alumnos, profesores, administradores del sistema y creadores de contenidos). Además, el sistema gestiona habitualmente los accesos, las actividades y los permisos del usuario y facilita herramientas de comunicación tanto síncronas como asíncronas.

Los elementos más característicos del modelo **SCORM** son:

- a) **Objeto de Contenido Compartido** (*Sharable Content Object*, **SCO** por sus siglas en inglés): es una colección de uno o más Activos (*Assets*), que representa la unidad mínima intercambiable entre sistemas compatibles con **SCORM**. Un **SCO** es el nivel de granularidad más bajo de un recurso de aprendizaje y es rastreado por un **LMS** usando el modelo de datos de **SCORM**. La única diferencia con entre un **SCO** y un Activo es que el **SCO** se comunica con un **LMS** usando una **Interfaz de Programación de Aplicaciones** (*Application Programming Interface*, **API** por sus siglas en inglés). Cumple con los requisitos de interoperabilidad, durabilidad y puede ser reutilizado.
- b) **Activos** (*Assets*): Recursos o elementos básicos, como archivos de texto, audio, video, etc. Estos recursos básicos se agrupan en los **SCOs**.
- c) **Ambiente en Tiempo de Ejecución** (*Run Time Environment*, **RTE** por sus siglas en inglés): Propone un ambiente estándar a través del cual los **SCOs** pueden intercambiar datos con el **LMS**. El **LMS** se encarga de enviar los contenidos al alumno y el contenido intercambia información sobre el alumno y el seguimiento de su interacción con el curso al **LMS**.

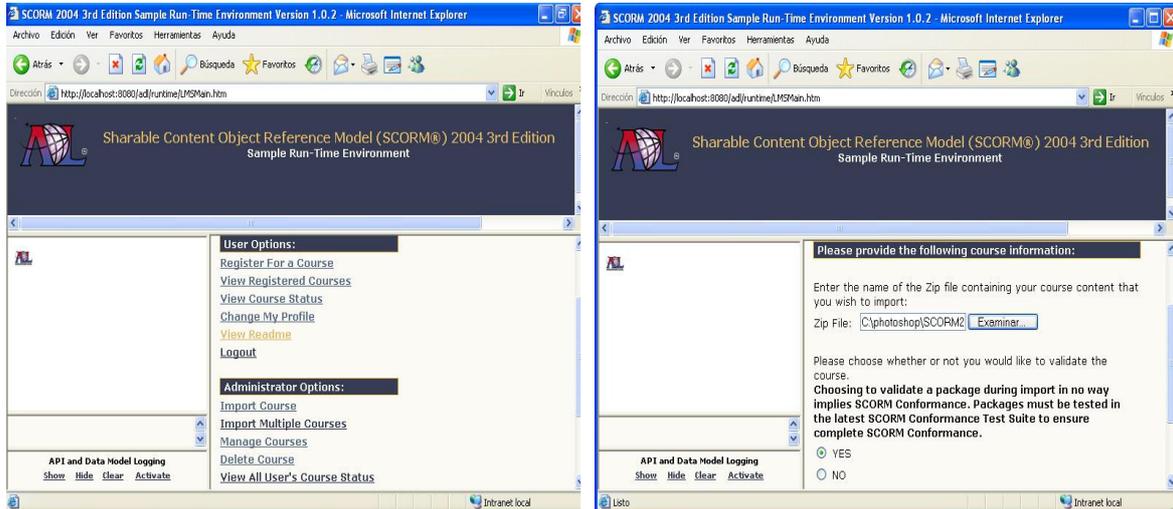
En la página principal de ADL se puede descargar el LMS de prueba: "SCORM 2004 3rd Edition Sample Run-Time Environment Version 1.0.2" [33].

2.3.4.1 Requerimientos de instalación "SCORM 2004 3rd Edition Sample Run-Time Environment Version 1.0.2"

Los requerimientos para instalar "SCORM 2004 3rd Edition Sample Run-Time Environment Version 1.0.2" son los siguientes:

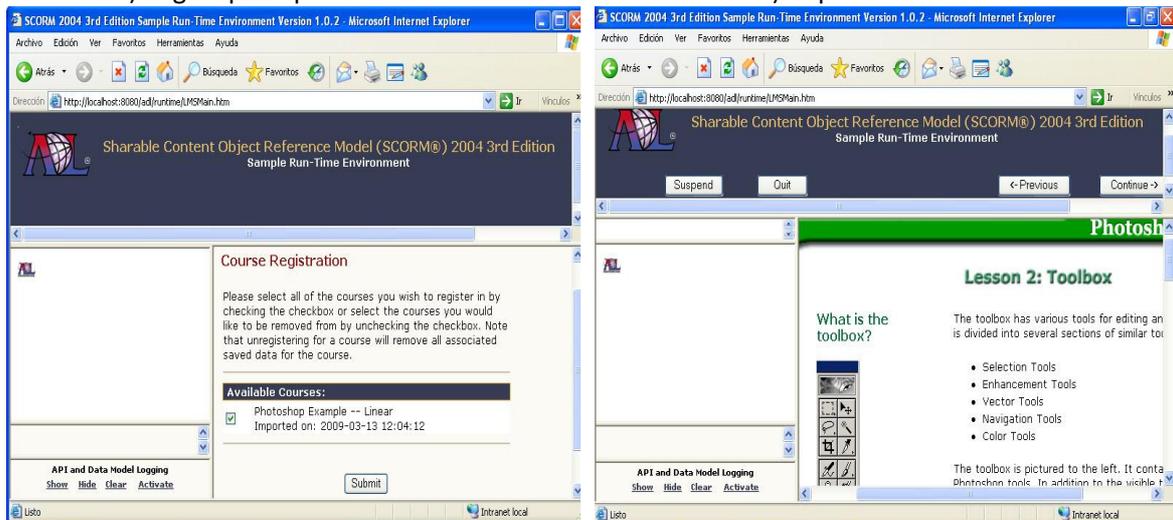
- Tener instalado el Kit de Desarrollo de Java (*Java Development Kit*, **JDK** por sus siglas en inglés) de cualquiera de las siguientes versiones: Java SE 6, J2SE 5.0 actualización 10 o posterior, J2SE 1.4.2_13 o posterior.
- Agregar la variable de entorno **JAVA_HOME** y la variable **PATH**.
- Después de la instalación es necesario configurar los archivos temporales de Internet dentro de Internet Explorer, a través del menú "Herramientas-> Opciones de Internet", en la pestaña "General", dar clic en el botón "Configuración" de la sección "Historial de Exploración" y por último seleccionar la opción: "Cada vez que visite la página *Web*".
- Asegurarse de que el plug-in de Java esté seleccionado como el ambiente Java por default para Internet Explorer, a través del Panel de Control, dar clic en el icono de "Java", en la pestaña "Avanzado", expandir la sección de "Java predeterminado para navegadores", seleccionar "Internet Explorer" y por último dar clic en "Aplicar".
- Permitir archivos "Java Script", para ello es necesario abrir el navegador Internet Explorer, en el menú "Herramientas-> Opciones de Internet", en la pestaña "Opciones Avanzadas", dentro de la sección "Seguridad", habilitar la opción "Permitir que el contenido activo se ejecute en los archivos de Mi Equipo".

El asistente de instalación de "SCORM 2004 3rd Edition Sample Run-Time Environment Version 1.0.2" permite su instalación, la **Figura 2.4 (a)** muestra la página principal de "SCORM 2004", en la cual es posible importar, eliminar y manejar cursos, así como registrarse y ver algún curso, entre otras opciones.



a) Página principal de SCORM

b) Importar un curso



c) Registrarse a un curso

d) Ver cursos registrados

Figura 2.4. "SCORM 2004 3rd Edition Sample RTE"

La **Figura 2.4 (b)** muestra cómo importar un curso, en este caso es necesario agregar un archivo ZIP, el cual contiene toda la estructura del curso, es decir, videos, imágenes, páginas HTML, así como el archivo "imsmanifest.xml", donde se especifica la estructura del curso, entre otros.

La **Figura 2.4 (c)** muestra cómo registrarse a un curso, a través de la opción "Registrarse a un curso (*Register For a Course*)" del menú principal.

Por último la **Figura 2.4 (d)** muestra el curso registrado, a través de la opción "Ver cursos registrados (*View Registered Courses*)".

2.3.4.2 Ventajas de SCORM

Las ventajas que proporciona el **SCORM** se presentan a continuación.

- **Interoperabilidad:** se refiere a la capacidad para transferir contenidos de aprendizaje desarrollados en un lugar, con un conjunto de herramientas o plataformas y poder usarlas, en otro lugar y con un conjunto distinto de herramientas o plataforma. Es decir, el mismo contenido debe funcionar de la misma forma cuando sea ejecutado en otro entorno SCORM compatible [34].
- **Portabilidad:** la capacidad para mover fácilmente contenidos desde un entorno a otro, donde debe funcionar, sin modificación ninguna, disponiendo simplemente de un navegador *Web* [34].
- **Accesibilidad:** la capacidad de localizar y recuperar elementos o piezas de aprendizaje desde un sitio remoto y poder distribuirlos a otros sitios remotos [34].
- **Reusabilidad:** la flexibilidad para incorporar piezas de aprendizaje en múltiples aplicaciones y contextos. El **SCORM** hace posible la elaboración de pequeñas piezas reutilizables que pueden ser recombinadas de diferentes maneras [34].
- **Durabilidad/Mantenibilidad:** la capacidad de resistir la evolución tecnológica y los cambios del contenido sin costes significativos de rediseño, recodificación o reconfiguración. El contenido debe durar lo suficiente para amortizar su coste y debe poder ser usando tanto tiempo como sea relevante [34].
- **Adaptabilidad:** la capacidad de personalizar e individualizar el aprendizaje a las necesidades de las personas y las organizaciones [34].

El **SCORM** maneja un modelo de datos para poder comunicar los **SCOs** con el **LMS**, a continuación se presenta dicho modelo.

2.3.4.3 Modelo de Datos SCORM 2004

Un modelo de datos es definido para poder mandar información específica a diferentes **LMS**. Por ejemplo si se requiere mandar información del resultado de la puntuación del alumno (*score*) como requisito, es necesario que el **LMS** sepa cómo manejarlo [35].

La **Figura 2.5** muestra la descripción del modelo de datos que utiliza **SCORM**, en el cual a través de un API y el modelo de datos, el **SCO** se puede comunicar con el **LMS**.

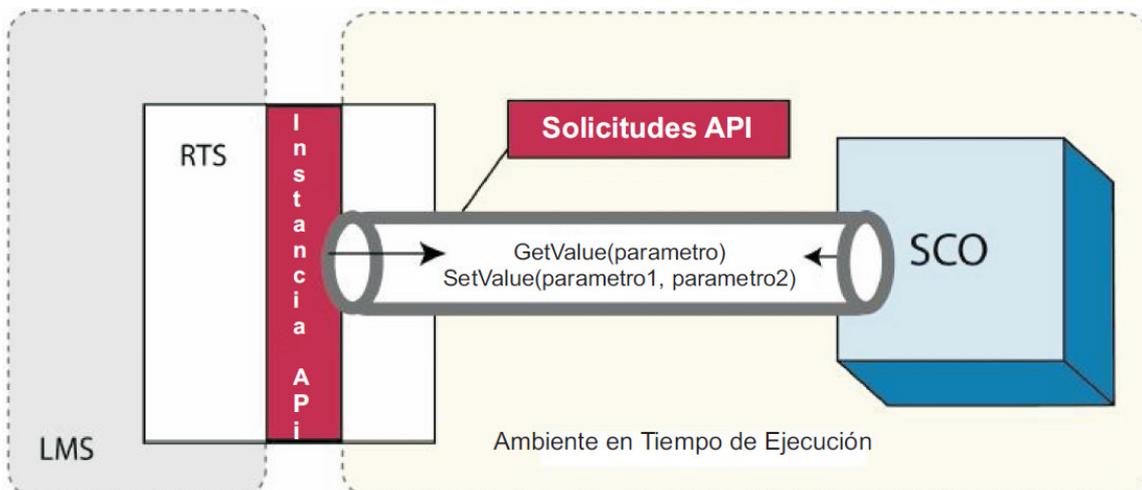


Figura 2.5. Uso del Modelo de Datos y el API

El modelo de datos está basado en el estándar IEEE LTSC CMI. 1484.11.1 que define un conjunto de elementos del modelo de datos que son necesarios para comunicar un objeto de contenido **SCO** con un **LMS**.

Este conjunto describe información acerca del estudiante, interacciones del estudiante con el **SCO**, información objetiva, estado de éxito y estado de finalización.

Para identificar el modelo de datos todos los elementos empiezan con *cmi*.

Los **SCO** están solamente obligados a usar la funciones API: *Initialize("")* y *Terminate("")*, para iniciar y terminar una sesión con el **LMS**.

Los siguientes elementos están definidos como colecciones de registros de datos: *cmi.comments_from_learner*, *cmi.comments_from_lms*, *cmi.objectives*, *cmi.interactions*.

El modelo de datos del ambiente en tiempo de ejecución de **SCORM** contiene un conjunto de elementos que pueden ser rastreados por el **SCO** con un **LMS** durante el tiempo de ejecución del **SCO**. Los elementos que pueden ser rastreados son: el estado, la puntuación, las interacciones, los objetivos, etc. Ciertos elementos tienen impacto en otros o son usados en coordinación con otros. Además algunos de estos elementos impactan el control y la secuencia de otros **SCOs**

En la Tabla 2.1 se describen algunos elementos del modelo de datos:

Tabla 2.1 Elementos del modelo de datos

Modelo de datos	Descripción
<i>cmi.comments_from_learner</i>	contiene los comentarios del estudiante
<i>cmi.comments_from_lms</i>	contiene comentarios y anotaciones que estarán disponibles para el estudiante
<i>cmi.completion_status</i>	indica si el estudiante ha completado el SCO
<i>cmi.completion_threshold</i>	identifica un valor que determina el progreso del alumno, puede ser útil para considerar si ha completado el SCO
<i>cmi.entry</i>	contiene información que afirma si el alumno ha accedido

	previamente al SCO
cmi.exit	indica cómo y porqué el alumno dejó el SCO
cmi.interactions	define información perteneciente a una interacción
cmi.launch_data	proporciona datos que el SCO puede utilizar para inicialización
cmi.learner_id	contiene información del identificador del estudiante
cmi.learner_name	representa el nombre del estudiante
cmi.learner_preference	define las preferencias del estudiante
cmi.location	representa una ubicación en el SCO
cmi.max_time_allowed	identifica el tiempo máximo en que el estudiante puede usar el SCO
cmi.mode	representa el modo en que será presentado al estudiante
cmi.objectives	identifica los objetivos de aprendizaje o desempeño del SCO
cmi.progress_measure	identifica la medida de progreso que ha hecho el estudiante
cmi.scaled_passing_score	identifica la puntuación para pasar un SCO
cmi.score	identifica la puntuación del estudiante
cmi.session_time	identifica la medida de tiempo que el estudiante ha tenido en la sesión actual
cmi.success_status	identifica si el alumno domina el SCO
cmi.time_limit_action	identifica qué debe hacer el SCO si el tiempo máximo permitido es excedido
cmi._version	es usado por el SCO para determinar la versión del modelo de datos soportado por el LMS

2.3.5 Salón Virtual basado en Web

El **Salón Virtual basado en Web**, es un Salón Virtual definido dentro de la Universidad Virtual (*Virtual University*). Un conjunto de agentes del mismo tipo, son responsables del manejo de diferentes usuarios y de sus peticiones [36].

El salón representa la interfaz entre los estudiantes y el profesor virtual y proporcionará materiales de aprendizaje personalizados.

En este contexto, los agentes son vistos como una comunidad de entidades de software colaborativas, móviles y autónomas, que cooperan entre ellas para realizar una tarea, en lugar de entidades distintas y separadas.

Diferentes tipos de agentes tienen una especialización única y por lo tanto encapsulan diferentes servicios, pero tienen características similares. Los agentes actúan de forma independiente, realizando su trabajo.

Los materiales de aprendizaje personalizado son creados solo si el sistema se adapta a las necesidades del estudiante en el proceso de interacción del estudiante. Ese proceso es soportado con la ayuda de varios agentes de diferentes especialidades, como por ejemplo un agente responsable de la clasificación del estudiante y un conjunto de agentes responsables de las

actividades de aprendizaje del estudiante, como encontrar el material relevante a las necesidades del estudiante.

En este sistema los agentes son usados como supervisores de los estudiantes. El rol y la necesidad de los agentes que soporten la creación y recuperación de la información acerca de las preferencias del estudiante así como datos personales como un modelo de aprendizaje está identificado en este sistema.

En el salón, los estudiantes se comunican con el Agente de Profesor Virtual (**APV**) responsable de proporcionar materiales de aprendizaje personalizados vía un conjunto de agentes de apoyo. La **Figura 2.6** muestra el modelo lógico del sistema.

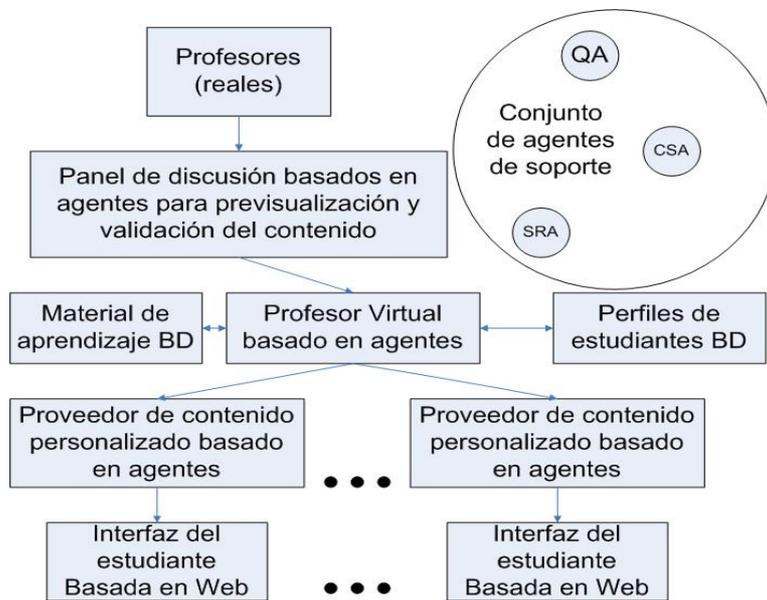


Figura 2.6. Modelo del Sistema basado en agentes

Cuando los estudiantes inician sesión en el sistema, el **APV** crea un Agente Proveedor de Contenido (**APC**). El **APC** es el responsable de presentar el material de aprendizaje a los estudiantes. Los materiales de aprendizaje pueden estar en cualquier formato. El **APV** se coordina con todos los **APC** y permite la comunicación con la Base de Datos del perfil del estudiante. La Base de Datos del perfil del estudiante contiene información personal de cada estudiante, así como sus motivaciones, conocimiento y deseos considerando el curso y la presentación del material de aprendizaje.

Los estudiantes se comunican con el **APV** directamente vía *Web* cuando asisten a un curso personalizado e interactúan indirectamente vía el correo electrónico cuando tienen preguntas acerca del contenido del curso.

Cuando el estudiante termina un curso, puede escoger realizar una prueba (*test*), en ese momento un mecanismo que extrae preguntas del material de aprendizaje se activa.

La **Figura 2.7** muestra la interfaz del sistema.

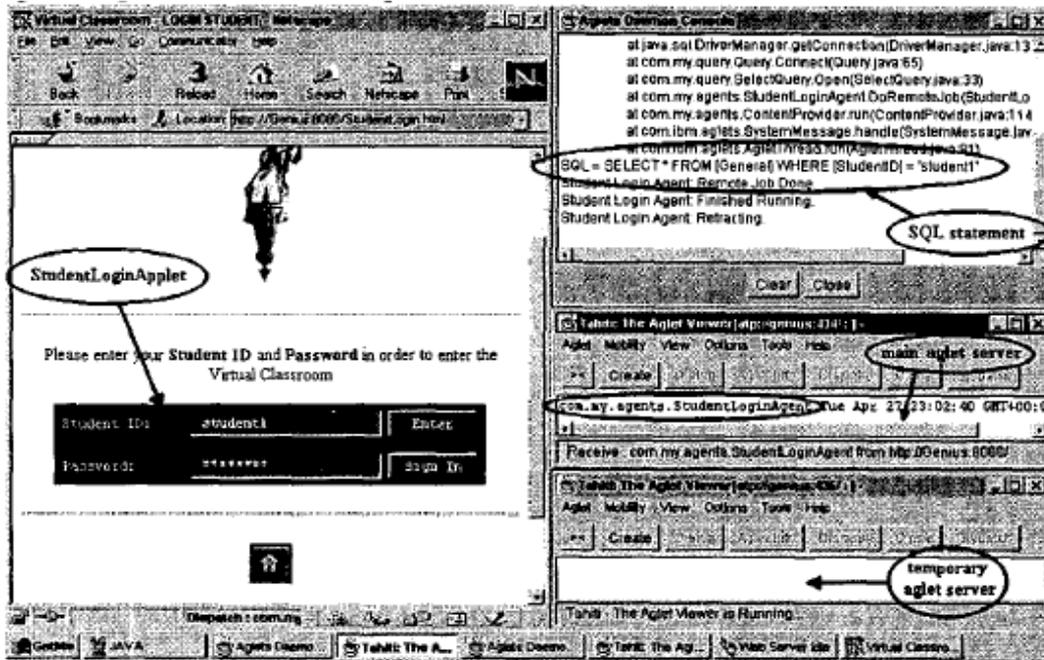


Figura 2.7. Sistema Virtual Classroom

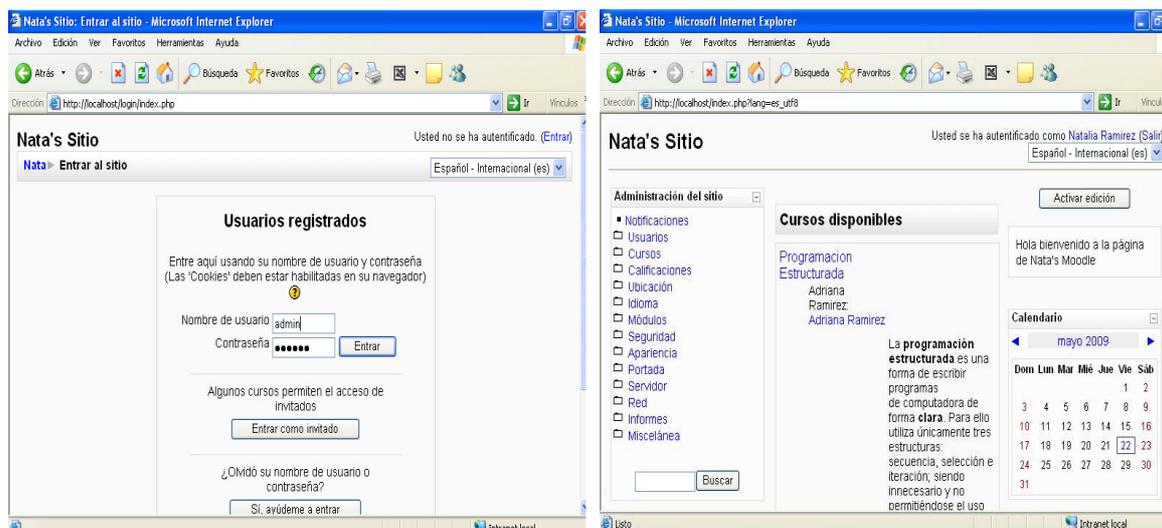
2.3.6 Moodle

El **Moodle** es un Sistema Administrador de Cursos (*Course Management System*, **CMS** por sus siglas en inglés) también conocido como **LMS**, de distribución libre. Se ha convertido en una herramienta en línea para los profesores, la cual les permite crear sitios *Web* dinámicos para sus alumnos. Para empezar a utilizarlo es necesario instalarlo en un servidor *Web* [37].

El **Moodle** se puede descargar en un paquete que contiene todo lo necesario para poder instalarlo o está la posibilidad de instalar cada herramienta que utiliza **Moodle** por separado (Apache, PHP, Base de Datos tipo SQL, etc.) [37].

Una vez instalado es posible iniciar el servicio a través del acceso directo *Start Moodle* y pararlo con *Stop Moodle*, los cuales se encuentran dentro del paquete descargado de **Moodle**. Al iniciar el servicio, se teclea la URL: <http://localhost> para entrar a la página principal de **Moodle**.

La **Figura 2.8 (a)** muestra la página principal de Login de **Moodle**, en la cual los usuarios dependiendo de su perfil: administrador, maestro y estudiante, pueden acceder al sistema.



a) Registro de usuarios

b) Página principal del administrador

Figura 2.8. Moodle perfil "Administrador"

La **Figura 2.8 (b)** muestra la página principal del administrador, en la cual el administrador puede agregar nuevos usuarios, crear nuevos cursos, configurar calificaciones, establecer un idioma, configurar los módulos, la apariencia, la seguridad, entre otros.

Los módulos principales de **Moodle**, a los cuales tiene acceso el profesor dentro de su página son los siguientes [38]:

Módulo de Tareas

- Los profesores pueden especificar la fecha final de entrega de una tarea y la calificación máxima que se le podrá asignar.
- Los estudiantes pueden subir sus tareas al servidor. Se registra la fecha en que se han subido.
- Para cada tarea en particular, puede evaluarse a los estudiante (calificaciones y comentarios).
- Las observaciones del profesor se adjuntan a la página de la tarea de cada estudiante y se le envía un mensaje de notificación.

Módulo de Consulta

- Puede utilizarse para votar sobre algún tema o para recibir una respuesta de cada estudiante.

Módulo Foro

- Hay diferentes tipos de foros disponibles: exclusivos para los profesores, de noticias del curso y abiertos a todos.

- El profesor puede obligar la suscripción de todos a un foro o permitir que cada persona elija a qué foros suscribirse de manera que se le envíe una copia de los mensajes por correo electrónico.
- El profesor puede elegir que no se permitan respuestas en un foro (por ejemplo, para crear un foro dedicado a anuncios).

Módulo Diario

- Los diarios constituyen información privada entre el estudiante y el profesor.
- Cada entrada en el diario puede estar motivada por una pregunta abierta.
- La clase entera puede ser evaluada, por cada entrada particular de diario.
- Los comentarios del profesor se adjuntan a la página de entrada del diario y se envía por correo la notificación.

Módulo Cuestionario

- Los profesores pueden definir una base de datos de preguntas que podrán ser reutilizadas en diferentes cuestionarios.
- El profesor puede determinar si los cuestionarios pueden ser resueltos varias veces y si se mostrarán o no las respuestas correctas y los comentarios
- Las preguntas y las respuestas de los cuestionarios pueden ser mezcladas (aleatoriamente).
- Las preguntas pueden crearse en HTML y con imágenes.

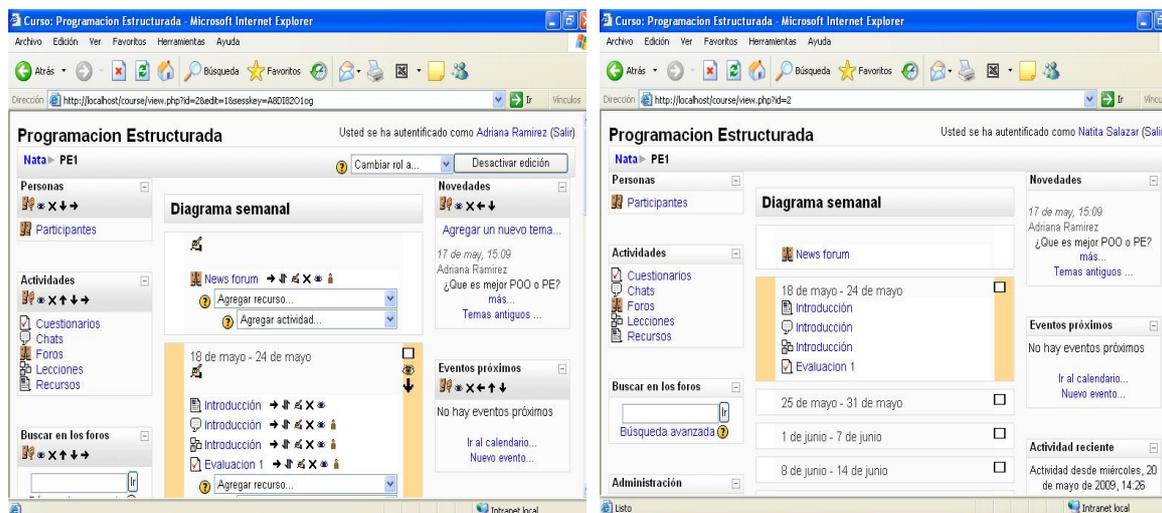
Módulo Recurso

- Admite la presentación de un importante número de contenido digital, Word, Powerpoint, Flash, vídeo, sonidos, etc.
- Los archivos pueden subirse y manejarse en el servidor, o pueden ser creados sobre la marcha usando formularios web (de texto o HTML).

Módulo Encuesta

- Los profesores pueden crear encuestas.
- Se pueden generar informes de las encuestas los cuales incluyen gráficos.
- A cada estudiante se le informa sobre sus resultados comparados con la media de la clase.

La **Figura 2.9 (a)** muestra la página principal del profesor, a través de la cual el profesor puede acceder a los módulos mencionados anteriormente.



a) Página principal del profesor

b) Página principal del estudiante

Figura 2.9. Moodle perfil "Profesor" y "Estudiante"

La **Figura 2.9 (b)** muestra la página principal del estudiante, el cual podrá acceder a los cursos registrados y asignados por el profesor, así como acceder a las actividades (chat, encuesta, cuestionario, lección, etc.) y recursos (página de texto, página *Web*, etc.) disponibles.

2.3.7 Sistema de Campus Virtual Politécnico

El Instituto Politécnico Nacional hace una propuesta alternativa, flexible e innovadora para ofertar servicios educativos, a la cual nombra **Sistema de Campus Virtual Politécnico (SCVP)**; éste requiere de esquemas de administración no tradicionales, sino generados a partir del análisis del funcionamiento de los procesos actuales. Ante la alta demanda de lugares para la educación media superior y superior, prevé elaborar nuevos programas de formación en el Campus Virtual [39].

Un Campus Virtual es un sistema académico con una estructura organizacional y procesos didácticos que hacen extensiva la oferta de actividades, cursos y programas educativos y que lleva a cabo las funciones de docencia, investigación, vinculación, extensión y difusión; desarrollado con base en las tecnologías de la información y las comunicaciones [39].

Entre sus características principales se tiene [39]:

- Modelos educativos innovadores de calidad y competitividad.
- Oferta educativa flexible y transcurricular.
- Desplazamiento de escenarios de baja tecnología hacia la alta tecnología.
- Utilización de entornos tecnológicos: redes locales, Intranet, Internet.
- Entornos de aprendizaje sustentados en ambientes colaborativos, en esquemas interinstitucionales y vinculados con dependencias externas.
- Los alumnos podrán estar exentos de permanecer físicamente en un Campus.
- La planta docente y administrativa deberá trabajar de forma no convencional, en los modos y contratos, ajustándose a este nuevo modelo educativo.

- Disponibilidad de una oferta educativa en línea, lo que impactará en una disminución de los ritmos de crecimiento de la infraestructura física.
- Cobertura geográfica.

La **Figura 2.10 (a)** muestra la página principal del **SCVP** [40], la cuál utiliza **Moodle**.

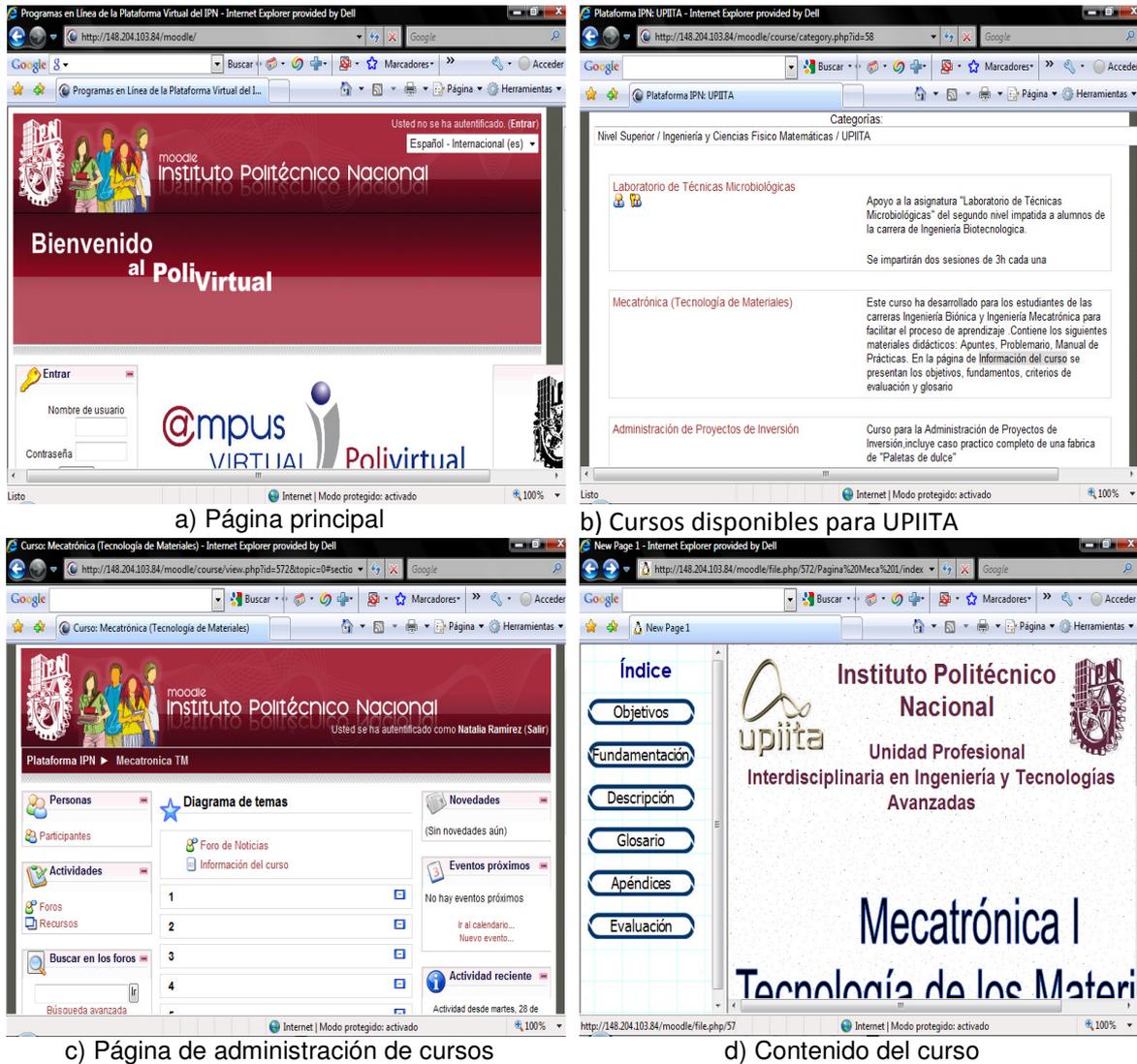


Figura 2.10. Sistema de Campus Virtual Politécnico

El **SCVP** muestra una lista de categorías de cursos clasificándolos a través del nivel de estudios, es decir, como primer categoría aparece el nivel medio superior y como segunda categoría el nivel superior, dentro de cada una aparecen las escuelas del Politécnico separadas por especialidad. La **Figura 2.10 (b)** muestra los cursos disponibles para la categoría de la escuela UPIITA, como se puede observar son muy pocos los cursos publicados, lo cual habla del poco seguimiento que se le ha dado a este proyecto.

En la **Figura 2.10 (c)** se muestra la página de administración de cursos de cada alumno, donde se muestran los temas disponibles, las calificaciones, los cursos registrados, entre otras cosas.

Por último, la **Figura 2.10 (d)** muestra el contenido del curso con interacción al contenido a través de los botones que se encuentran en la barra lateral izquierda.

2.4 COMPONENTES DE SOFTWARE

Clemens Szyperski, considerado el creador de esta tecnología da la siguiente definición: “*Un **Componente de Software** es una unidad de composición con interfaces especificadas contractualmente y dependencias de contexto explícitas. Un componente de software puede desplegarse independientemente y está sujeto a la composición por terceras partes*” [41].

Las propiedades características de un componente son:

- Es una unidad de despliegue independiente.
- Es una unidad de composición de terceras partes.
- No tiene un estado observable (externamente).

A diferencia de los objetos, los cuales tienen las siguientes características:

- Es una unidad de instanciación, tiene una identidad única.
- Puede tener estado y puede ser observable (externamente).
- Encapsula su estado y su comportamiento.

Los **componentes de software** están hechos para composición, que permite que componentes prefabricados sean reutilizados, reordenándolos en cada nueva composición. Para ser específicos los **componentes de software** son unidades independientes de producción, adquisición y despliegue, que pueden ser integrados en un sistema.

Existen dos opciones en el desarrollo tradicional de software, que son hacerlo o comprarlo, estos dos enfoques pueden resultar demasiado caros, el concepto de **componente de software** representa un camino intermedio que podría resolver este problema.

Cada **componente** es un producto estandarizado, pero el proceso de montaje de componentes permite una personalización importante. Es por ello que los **componentes** de diferentes calidades y características (nivel de rendimiento, eficiencia de recursos, solidez, grado de certificación, etc.) podrían estar disponibles a diferentes precios [42].

2.5 AGENTES INTELIGENTES

Actualmente no hay una definición estándar de lo que es la computación basada en agentes o el término de agentes inteligentes, debido a ello una definición aceptada es la siguiente:

*“Un **agente** es un sistema de cómputo situado en algún ambiente, el cual es capaz de tomar acciones autónomas en dicho ambiente con el fin de alcanzar sus objetivos”* [43].

Cuando se habla de autonomía se refiere principalmente a que el sistema debe ser capaz de actuar sin la intervención directa de los humanos o de otros agentes y además debe de tener cierto control sobre sus acciones y su estado interno, a diferencia de los objetos en los sistemas orientados a objetos, los cuales no tienen control sobre sus propias acciones.

Las características de los agentes inteligentes son las siguientes:

2.5.1 Características de los agentes inteligentes

- **Responsivo:** Los agentes deben de percibir su ambiente (el cual puede ser un ambiente físico, un usuario, una colección de agentes, Internet, etc.) y responder a cambios que ocurran en él [43].
- **Proactivo:** Los agentes no deben solamente actuar en respuesta a su ambiente, deben también ser oportunistas, tener un comportamiento dirigido a sus objetivos y tomar iniciativas en caso de ser apropiado [43].
- **Social:** Los agentes deben de ser capaces de interactuar cuando lo crean conveniente, con otros agentes artificiales y humanos a fin de completar la solución a sus problemas y para ayudar a los demás con sus actividades [43].

Un sistema basado en agentes es aquel en el que la clave de abstracción usada es un agente, y debe de ser conceptualizado en términos de agentes.

2.6 ARQUITECTURA IEEE 1484 - LTSA

El comité de estándares de la tecnología de aprendizaje del IEEE 1484 propuso una arquitectura que se pretende sea un estándar para todos los sistemas de tecnologías de aprendizaje. Cada una de las capas se describe a continuación [44]:

- **Las interacciones del ambiente y de los estudiantes:** referente a la adquisición, transferencia, intercambio, formulación, etc., del conocimiento y/o información a través de la interacción con el ambiente [44].
- **Características del diseño relacionadas al estudiante:** referentes a los efectos que tienen sobre el diseño de los sistemas de tecnología de aprendizaje [44].
- **Sistema de componentes:** describe la arquitectura basada en componentes, identificando los procesos que intervienen en el sistema y se centran en el estudiante [44].
- **Implementación de perspectivas y prioridades:** describe los sistemas tecnológicos de aprendizaje desde una amplia variedad de perspectivas por referencia a subconjuntos de la capa de componentes del sistema [44].

— **Interoperabilidad y funcionalidad de los componentes-codificación, APIs, protocolos:** describen los componentes genéricos interoperables e interfaces de una arquitectura tecnológica de aprendizaje basada en las tecnologías de la información [44]. La **Figura 2.11** muestra la arquitectura IEEE 1484 – LTSA con algunas mejoras en la capa del sistema de componentes [44]:

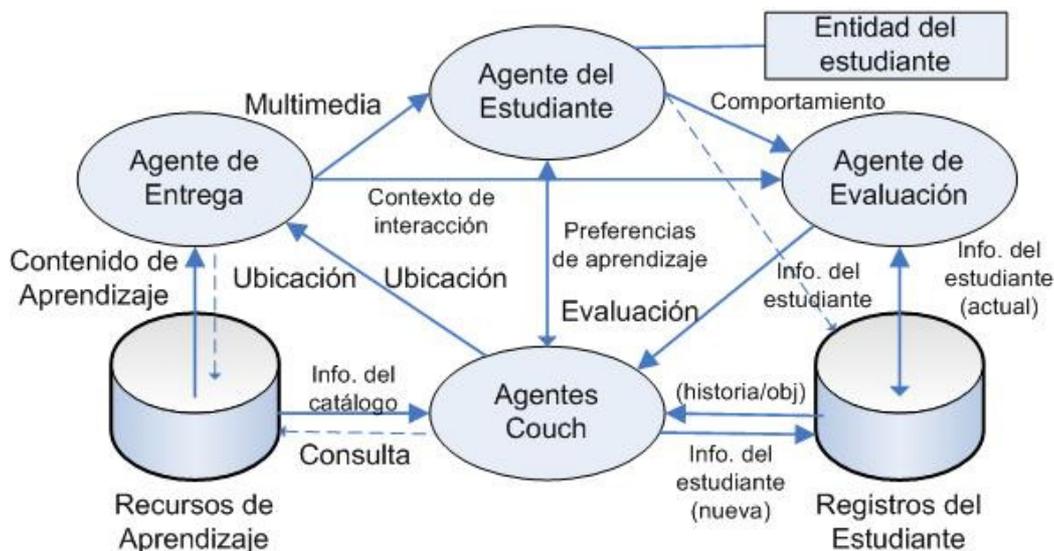


Figura 2.11. Arquitectura IEEE 1484 – LTSA

2.7 RESUMEN

En el presente capítulo se trataron temas importantes dentro del marco teórico para el desarrollo del trabajo de tesis como lo es la **WBE**, también se habló de los principales sistemas de entrenamiento que se encuentran en la *Web* y se hizo una tabla comparativa para remarcar la importancia del desarrollo planteado en esta investigación.

Debido a la falta de herramientas que permitan automatizar la generación de cursos de entrenamiento, se propone el desarrollo de **SEVirtual**, ya que es una herramienta que hereda las ventajas de la programación basada en componentes (las cuales se explican a continuación), los costos de licenciamiento son muy bajos, implementa el uso de estándares internacionales y algo muy importante, cuenta con un sistema asistencial basado en agentes, el cual permitirá una retroalimentación más inteligente para la parte de evaluación del alumno, recomendaciones de ejercicios, libros y ligas de Internet sobre los temas que se le hayan dificultado más al alumno así como la reconfiguración del curso en base también a dichos temas.

En el capítulo siguiente se discutirá el análisis y diseño del sistema de entrenamiento virtual basado en *Web*.

CAPÍTULO 3

ANÁLISIS Y DISEÑO DE SEVIRTUAL

En el presente capítulo se aborda el análisis y diseño de SEVirtual en el cual se muestra el modelado del sistema haciendo uso del **Lenguaje Unificado de Modelado** (*Unified Modeling Language*, UML por sus siglas en inglés) [50-53], el cual es un estándar ampliamente utilizado para la construcción de modelos orientados a objetos, para ello se muestra el diagrama general de casos de uso [51], los diagramas de secuencias principales [51] así como el diagrama de clases [51] y de paquetes [51]. Para el modelado de la base de datos se muestra el diagrama relacional.

3.1 DIAGRAMA DE CASOS DE USO Y SECUENCIA

Tomando en cuenta los requerimientos del sistema SEVirtual se procede al análisis y diseño del mismo, a través de los diagramas de casos de uso y de secuencia para cada escenario principal del sistema, que a continuación se presentan: el diagrama de casos de uso general del sistema, el diagrama de casos de uso del administrador, el diagrama de casos de uso del profesor y el diagrama de casos de uso del alumno. Para lo cual se identificaron tres usuarios principales:

- Administrador: Persona encargada de llevar el control de alta, baja y cambios de usuarios.
- Profesor: Usuario autorizado para llevar la administración de sus cursos.
- Alumno: Persona autorizada para visualizar los cursos, utilizar los recursos, evaluar su conocimiento y recibir una retroalimentación inteligente.

3.1.1 Diagrama de casos de uso general del sistema

Cualquiera de los tres tipos de usuario mencionados identificados para el sistema pueden iniciar sesión para poder acceder a sus actividades introduciendo su correo y contraseña y para finalizar se les permite cerrar su sesión. La **Figura 3.1** muestra el diagrama de casos de uso general del sistema, en el cual se encuentran los actores y módulos principales.

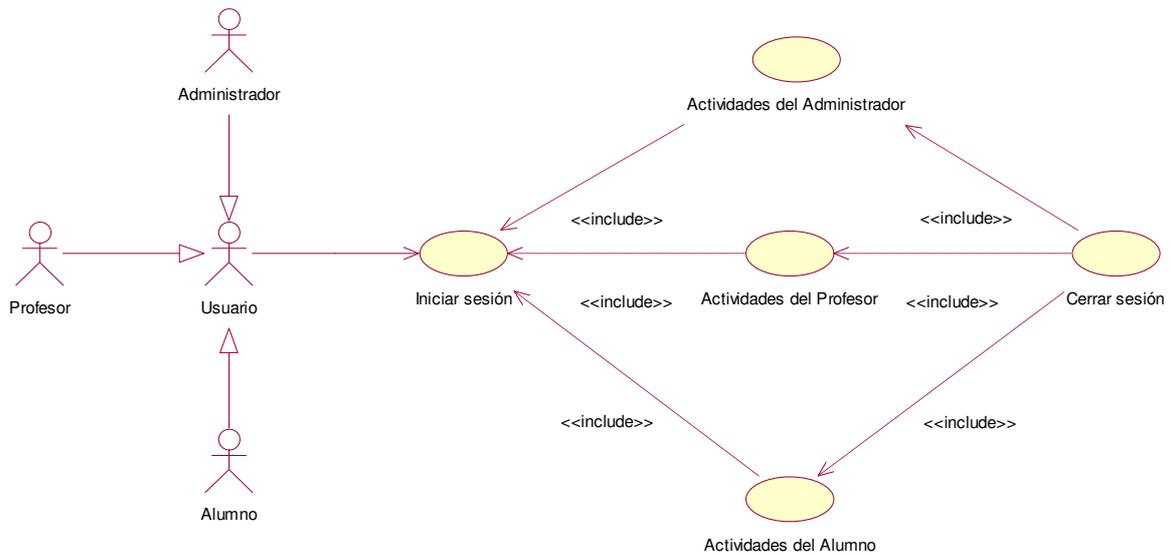


Figura 3.1 Diagrama del caso de uso general del sistema

3.1.2 Análisis del caso de uso general del sistema

La **Tabla 3.1** muestra el análisis del caso de uso: general del sistema, teniendo como actores al alumno, profesor y administrador.

Tabla 3.1 Análisis del caso de uso general del sistema

Caso de uso	
Nombre: Caso de uso general del sistema	
Actores: Alumno, profesor, administrador	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. Este caso de uso inicia cuando el usuario introduce el correo y el <i>password</i> para iniciar la sesión.	2. El Sistema valida si el usuario está registrado y si coincide el <i>password</i> , guarda la sesión y le muestra la parte del sistema que le corresponde de acuerdo a sus privilegios.
3. El usuario realiza las actividades que requiere.	4. El Sistema responde a la solicitud de dichas actividades.
5. El usuario cierra la sesión.	6. El Sistema remueve la sesión del usuario.
Cursos alternativos	
Línea 2. El usuario y/o <i>password</i> son incorrectos, no se guarda la sesión y se muestra la advertencia.	

3.1.2.1 Diagrama de secuencia del caso de uso: Iniciar sesión

En la **Figura 3.2** se muestra la interacción del usuario con el sistema, cuando éste desea iniciar sesión, primero el usuario envía el correo y el *password*, los cuales son necesarios para validar si el usuario existe y si su *password* es correcto, una vez validado se crea su objeto correspondiente y se guarda en la sesión, se obtiene el tipo y se envía al cliente para que cargue el menú correspondiente a las actividades de dicho usuario.

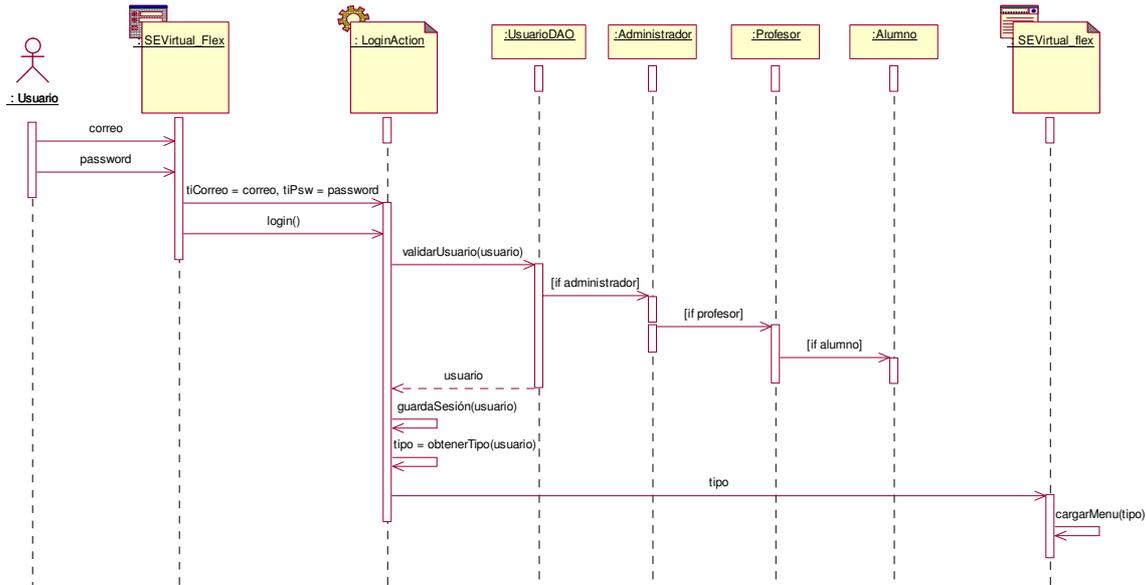


Figura 3.2 Diagrama de secuencia del caso de uso: Iniciar sesión

3.1.3 Diagrama de casos de uso del administrador

La **Figura 3.3** muestra el diagrama de casos de uso del administrador, en el cual se muestran sus actividades principales. Las cuales una vez iniciada su sesión son: dar de alta usuarios, borrar usuarios y modificar usuarios. Dentro de la opción dar de alta usuario, el administrador puede introducir los datos para el nuevo usuario y darlo de alta, el administrador también puede borrar algún usuario buscándolo primero en el sistema, así como modificar los datos de algún usuario.

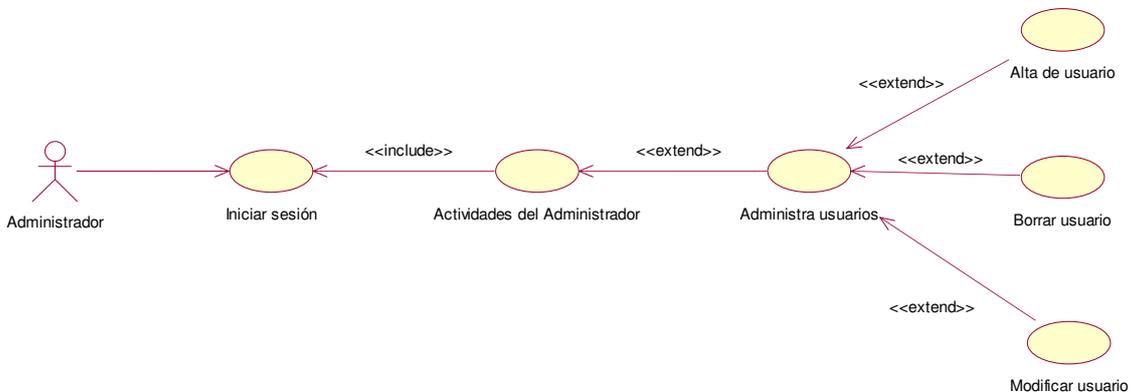


Figura 3.3 Diagrama de casos de uso del administrador

3.1.4 Análisis del caso de uso del administrador

La **Tabla 3.2** muestra el análisis del caso de uso: general del administrador, teniendo como actividades principales: dar de alta usuarios, borrar usuarios y modificar usuarios.

Tabla 3.2 Análisis del caso de uso del administrador

Caso de uso	
Nombre: Caso de uso del administrador Actores: Administrador	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El administrador inicia sesión.	2. El sistema muestra el menú y la parte del sistema que le corresponde al administrador.
3. El administrador selecciona una de las actividades que desea realizar.	4. El Sistema muestra el ambiente correspondiente a esa actividad.

La **Tabla 3.3** muestra el análisis del caso de uso: alta de usuario, teniendo como actor al administrador.

Tabla 3.3 Análisis del caso de uso: Alta de usuario

Caso de uso	
Nombre: Alta de usuario Actores: Administrador	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El administrador selecciona la opción: Alta de usuario.	2. El sistema muestra el formulario correspondiente a llenar con los datos del nuevo usuario.
3. El administrador ingresa los datos del nuevo usuario como son: correo, <i>password</i> , nombre y edad.	4. El sistema verifica si el usuario no existe y si no existe lo registra, una vez registrado envía un mensaje al usuario con el resultado de su operación.
Cursos alternativos	
Línea 4. El usuario ya existe por lo que se le pide al administrador ingresar otro nombre de usuario.	

3.1.4.1 Diagrama de secuencia del caso de uso: Alta de usuario

En la **Figura 3.4** se muestra la interacción con el sistema, cuando el administrador desea dar de alta un usuario, para ello primero envía los datos del nuevo usuario necesarios para registrarlo, posteriormente el sistema crea el objeto correspondiente al tipo de usuario y verifica si el usuario no existe, si éste no existe lo registra.

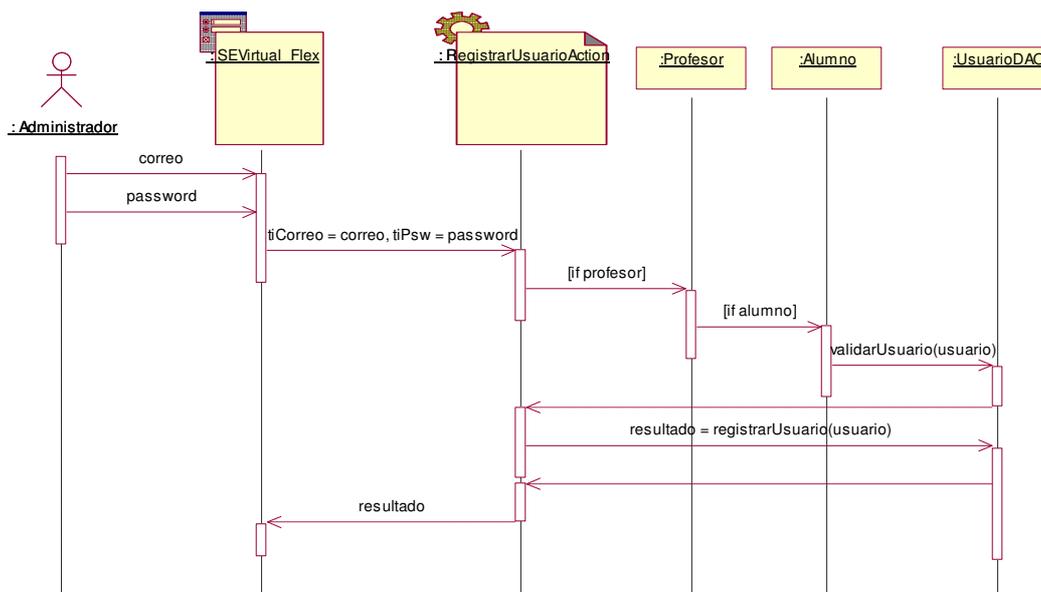


Figura 3.4 Diagrama de secuencia del caso de uso: Alta de usuario

La **Tabla 3.4** muestra el análisis del caso de uso: borrar usuario, teniendo como actor al administrador.

Tabla 3.4 Análisis del caso de uso: Borrar usuario

Caso de uso	
Nombre: Borrar usuario Actores: Administrador	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El administrador selecciona la opción: Borrar usuario.	2. El Sistema muestra el formulario donde se desplegarán los datos del usuario a borrar como son: correo, password, nombre, edad y tipo de usuario.
3. El administrador introduce el correo de usuario a buscar.	4. El Sistema busca si existe el usuario y si existe despliega sus datos.
5. El administrador selecciona la opción: borrar.	6. El Sistema recupera el correo del usuario, lo borra y muestra un mensaje con el resultado de la operación.
Cursos alternativos	

Línea 4. El usuario no existe, por lo cual el Sistema le avisa al usuario.
Línea 6. No se puede borrar al usuario.

3.1.4.2 Diagrama de secuencia del caso de uso: Borrar usuario

En la **Figura 3.5** se muestra la secuencia necesaria para borrar un usuario del sistema, primero el administrador introduce el correo del usuario a borrar, el sistema busca al usuario y si existe despliega sus datos, por último el usuario selecciona la opción borrar y el sistema lo borra.

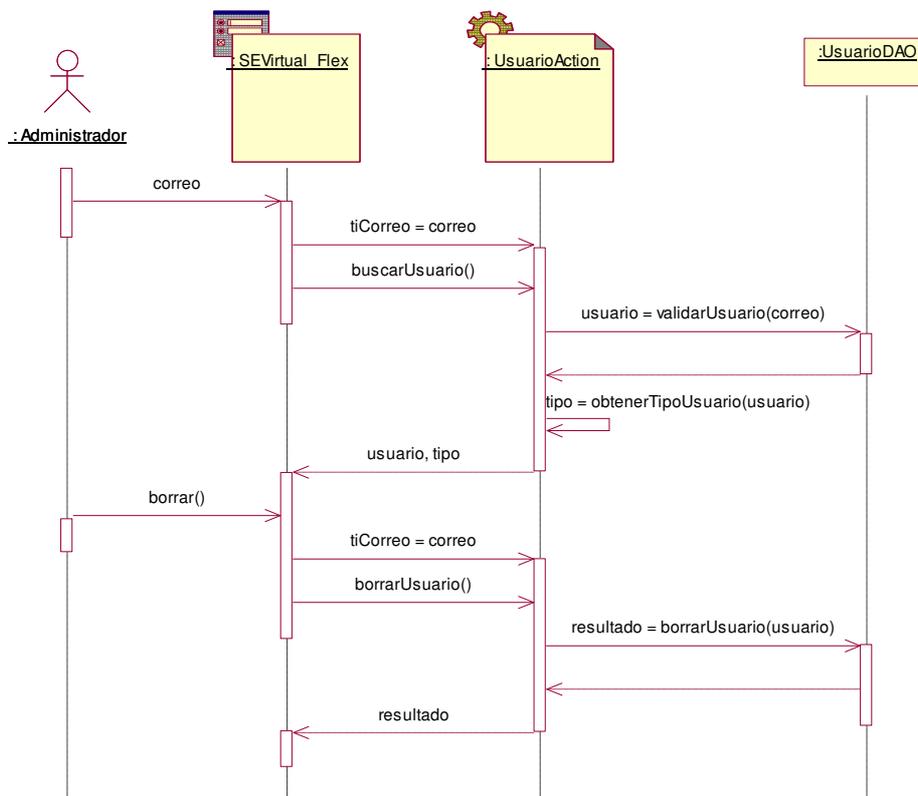


Figura 3.5 Diagrama de secuencia del caso de uso: Borrar usuario

La **Tabla 3.5** muestra el análisis del caso de uso: modificar usuario, teniendo como actor al administrador.

Tabla 3.5 Análisis del caso de uso: Modificar usuario

Caso de uso	
Nombre: Modificar usuario	
Actores: Administrador	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema

1. El administrador selecciona la opción: Modificar usuario.	2. El Sistema muestra el formulario donde se desplegarán los datos del usuario a modificar como son: correo, <i>password</i> , nombre, edad y tipo de usuario.
3. El administrador introduce el correo de usuario a buscar.	4. El Sistema busca al usuario y despliega sus datos.
5. El administrador edita los datos del usuario y selecciona la opción: actualizar.	6. El Sistema actualiza todos los datos del usuario y muestra un mensaje con el resultado de la operación.
Cursos alternativos	
Línea 4. El usuario no existe, por lo cual el Sistema le avisa al usuario.	
Línea 6. No se puede actualizar al usuario.	

3.1.4.3 Diagrama de secuencia del caso de uso: Modificar usuario

En la **Figura 3.6** se muestra la interacción con el sistema, cuando el administrador desea modificar los datos de un usuario, primero el administrador introduce el correo, el sistema busca al usuario y el administrador edita los datos de usuario.

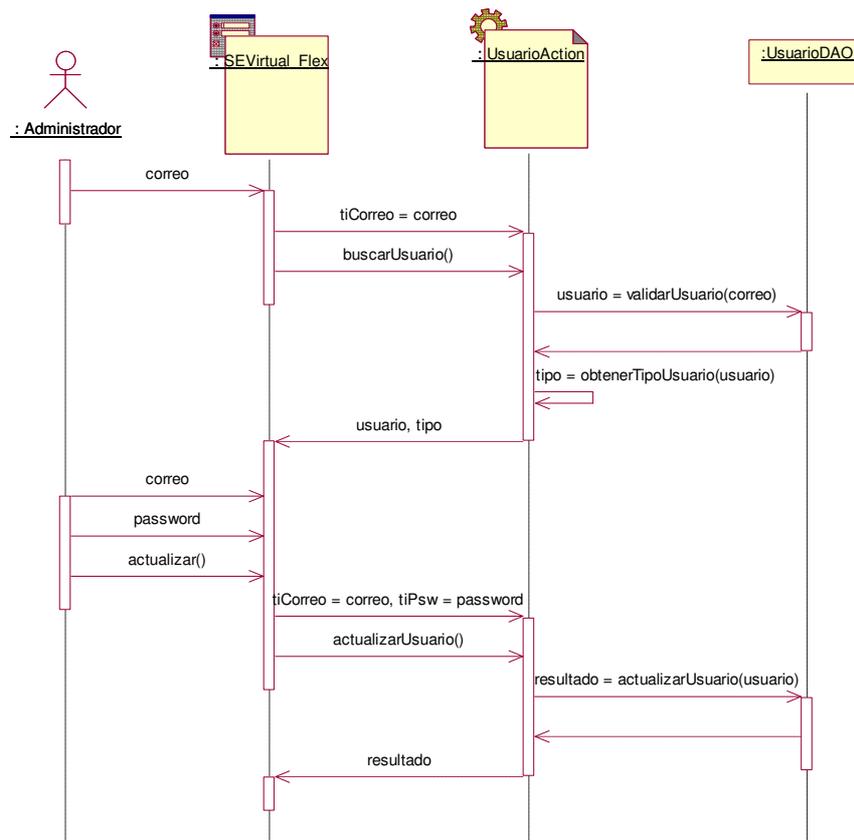


Figura 3.6 Diagrama de secuencia del caso de uso: Modificar usuario

3.1.5 Diagrama de casos de uso del profesor

La **Figura 3.7** muestra el diagrama de casos de uso del profesor, en el cual se muestran sus actividades principales. Las cuales una vez iniciada su sesión son: crear un curso, ver los cursos creados, registrar curso, editar curso, previsualizar curso y eliminar curso. Dentro de la opción ver los cursos creados, el profesor puede obtener un listado de todos los cursos que ha creado, así como seleccionar alguno para editarlo, previsualizarlo o eliminarlo, el sistema se encargará de cargar el ambiente correspondiente a la opción seleccionada por el profesor. Además el profesor también puede crear un curso nuevo con un estructurado abierto, así como seleccionar componentes prefabricados donde insertará el contenido del curso y la opción de subir multimedia al mismo.

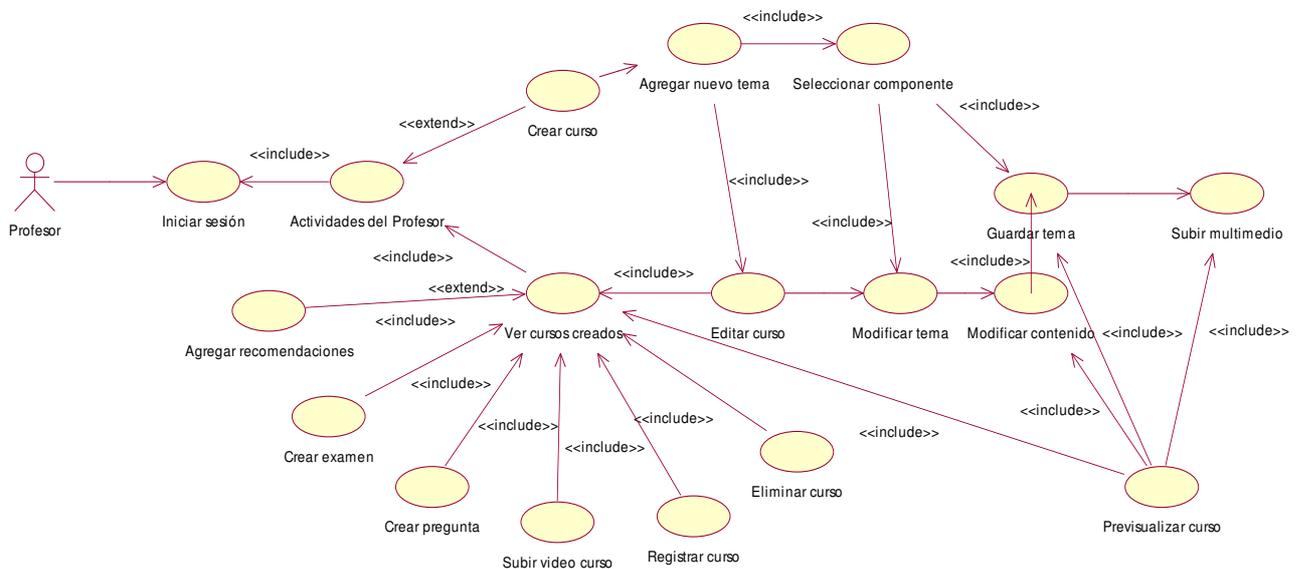


Figura 3.7 Diagrama de casos de uso del profesor

La **Tabla 3.6** muestra el análisis del caso de uso: general del profesor, teniendo como actividades principales: crear curso, ver los cursos creados, editar, eliminar, registrar y previsualizar un curso.

3.1.6 Análisis del caso de uso del profesor

Tabla 3.6 Análisis del caso de uso del profesor

Caso de uso	
Nombre: Caso de uso del profesor	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor inicia sesión.	2. El sistema muestra el menú y la parte del sistema que le corresponde al profesor.
3. El profesor selecciona una de las	4. El Sistema muestra el ambiente

actividades que desea realizar.	correspondiente a esa actividad.
---------------------------------	----------------------------------

La **Tabla 3.7** muestra el análisis del caso de uso: crear curso, teniendo como actor al profesor.

Tabla 3.7 Análisis del caso de uso: Crear curso

Caso de uso	
Nombre: Crear curso	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona la opción: Crear nuevo curso.	2. El Sistema pide el nombre del curso.
3. El profesor introduce el nombre del curso.	4. El Sistema crea la estructura del curso, basada en metadatos.

3.1.6.1 Diagrama de secuencia del caso de uso: Crear curso

En la **Figura 3.8** se muestra la interacción con el sistema, cuando el profesor desea crear un curso, para ello el profesor introduce el nombre del curso y el sistema se encarga de crear la estructura de metadatos con el nombre del curso.

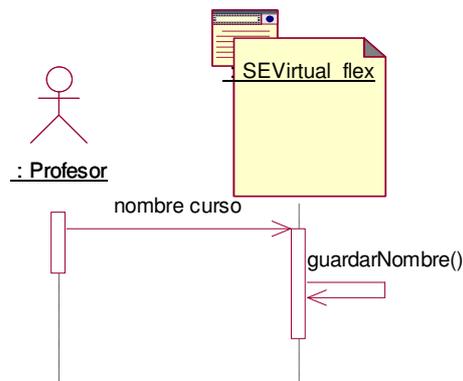


Figura 3.8 Diagrama de secuencia del caso de uso: Crear curso

La **Tabla 3.8** muestra el análisis del caso de uso: agregar nuevo tema, teniendo como actor al profesor.

Tabla 3.8 Análisis del caso de uso: Agregar nuevo tema

Caso de uso	
Nombre: Agregar nuevo tema	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema

1. El profesor agrega un nuevo tema, seleccionando la opción: agregar tema.	2. El Sistema pide el nombre del tema.
3. El profesor introduce el nombre del tema.	4. El Sistema agrega el tema a la estructura de metadatos del curso.

3.1.6.2 Diagrama de secuencia del caso de uso: Agregar nuevo tema

En la **Figura 3.9** se muestra la interacción con el sistema, cuando el profesor desea agregar un nuevo tema, para ello introduce el nombre del tema y el sistema se encarga de actualizar la estructura de metadatos del curso.

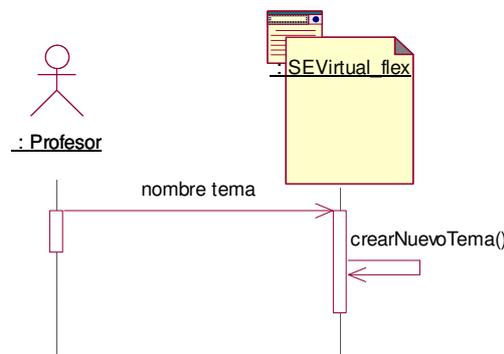


Figura 3.9 Diagrama de secuencia del caso de uso: Agregar nuevo tema

La **Tabla 3.9** muestra el análisis del caso de uso: seleccionar componente, teniendo como actor al profesor.

Tabla 3.9 Análisis del caso de uso: Seleccionar componente

Caso de uso	
Nombre: Seleccionar componente	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona uno de los componentes predefinidos para agregar el contenido del curso.	2. El Sistema muestra el componente a través del cual el profesor podrá insertar el contenido.
3. El profesor introduce el contenido del tema.	
Cursos alternativos	
Línea 1. El profesor puede agregar más temas. El profesor puede seleccionar entre los siguientes componentes: solo texto, titulo texto, titulo subtítulo, titulo doble texto o solo título.	

3.1.6.3 Diagrama de secuencia del caso de uso: Seleccionar componente

En la **Figura 3.10** se muestra la interacción con el sistema, cuando el profesor selecciona un componente para introducir el contenido del tema, una vez introducido el tema el profesor podrá guardarlo.

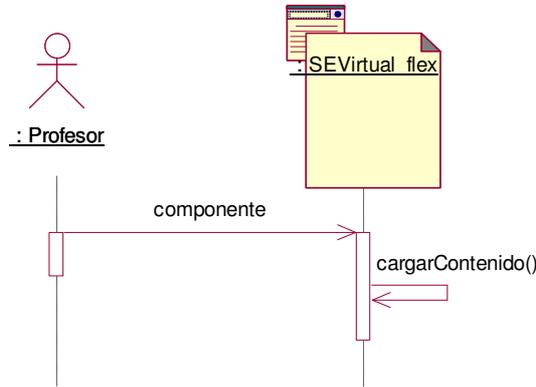


Figura 3.10 Diagrama de secuencia del caso de uso: Seleccionar componente

La **Tabla 3.10** muestra el análisis del caso de uso: guardar tema, teniendo como actor al profesor.

Tabla 3.10 Análisis del caso de uso: Guardar tema

Caso de uso	
Nombre: Guardar tema	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor guarda el tema una vez seleccionado el componente e insertado el contenido.	2. El Sistema verifica si el curso existe a través de la ruta, si éste no existe entonces el Sistema lo registra y de cualquier manera escribe el archivo XML en el servidor en la ruta especificada. Por último el Sistema despliega el resultado de la operación.

3.1.6.4 Diagrama de secuencia del caso de uso: Guardar tema

En la **Figura 3.11** se muestra la interacción con el sistema, cuando el profesor desea guardar un tema, para ello el profesor habrá insertado el contenido en el componente y el sistema se encargará de guardarlo en el servidor.

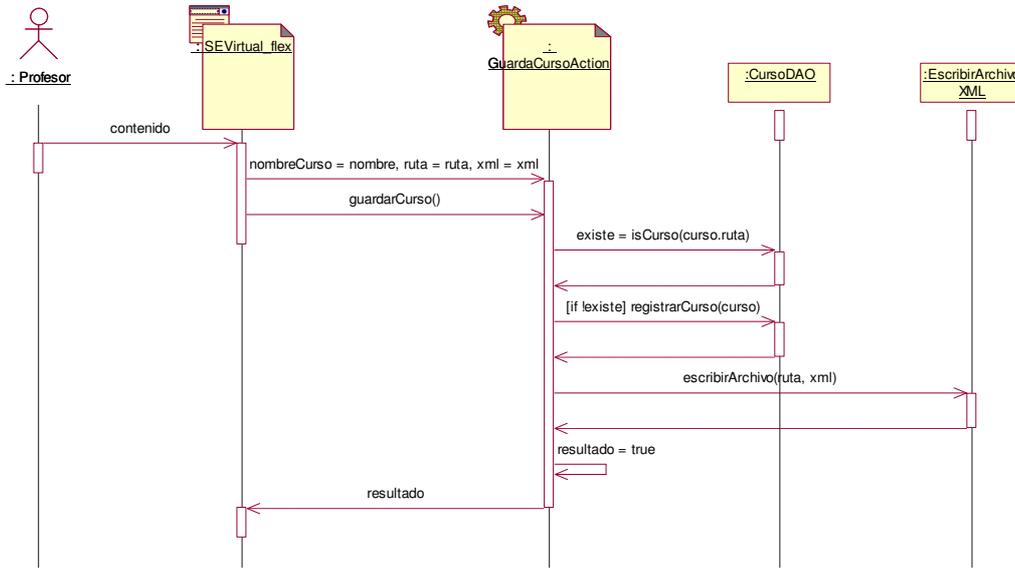


Figura 3.11 Diagrama de secuencia del caso de uso: Guardar tema

La **Tabla 3.11** muestra el análisis del caso uso: subir multimedia, teniendo como actor al profesor.

Tabla 3.11 Análisis del caso de uso: Subir multimedia

Caso de uso	
Nombre: Subir multimedia	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona la opción: subir multimedia.	2. El Sistema carga la ventana que permita al profesor seleccionar un archivo.
1. El profesor selecciona el nombre del archivo multimedia, con los siguientes formatos: mp3, jpg, gif, png y flv.	2. El Sistema envía el multimedia al servidor, crea los directorios y guarda el archivo con el nombre original, posteriormente verifica si el curso existe en la base de datos, si no existe lo registra y de cualquier manera escribe el archivo en la ruta especificada en el servidor.
Cursos alternativos	
Línea 1. El profesor puede previsualizar el curso.	

3.1.6.5 Diagrama de secuencia del caso de uso: Subir multimedia

En la **Figura 3.12** se muestra la interacción con el sistema, cuando el profesor desea subir un multimedia al curso, para ello primero selecciona el nombre del archivo y posteriormente el sistema se encarga de guardarlo en el servidor.

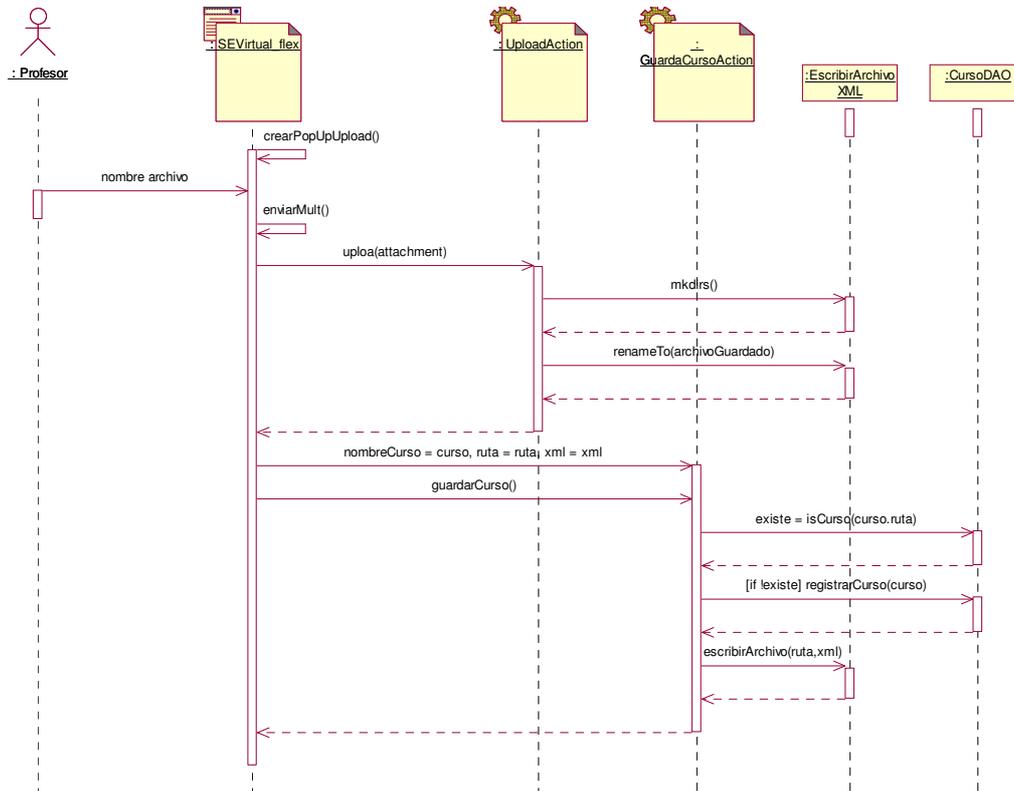


Figura 3.12 Diagrama de secuencia del caso de uso: Subir multimedia

La **Tabla 3.12** muestra el análisis del caso de uso: previsualizar curso, teniendo como actor al profesor.

Tabla 3.12 Análisis del caso de uso: Previsualizar curso

Caso de uso	
Nombre: Previsualizar curso	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema

1. El profesor selecciona la opción: previsualizar curso.	2. El Sistema muestra el ambiente del curso creado por el profesor, para ello carga la estructura de metadatos del servidor, llena los componentes con la información del curso y carga el tema del índice cero, es decir, el primer tema.
3. El profesor navega por el curso a través del árbol de la estructura del curso o a través de la barra de navegación, el profesor puede seleccionar la opción ir al principio.	4. El Sistema actualiza el índice en la posición cero para cargar el primer tema.
5. El profesor navega por el curso a través de la opción ir al anterior.	6. El Sistema actualiza el índice disminuyéndolo en una unidad y carga el contenido del tema.
7. El profesor navega por el curso a través de la opción ir al siguiente.	8. El Sistema actualiza el índice incrementándolo en una unidad y carga el contenido del tema.
9. El profesor navega por el curso a través de la opción ir al final.	10. El Sistema actualiza el índice en la posición final y carga el contenido del tema.
11. El profesor selecciona la opción: reproducir para comenzar la reproducción del video explicativo del curso.	12. El Sistema comienza la reproducción e inicia el contador para cambiar automáticamente los temas de acuerdo a la reproducción del video.
13. El profesor selecciona un tema y escoge la opción: multimedia para visualizar los multimedios de ese tema.	14. El Sistema carga el componente que muestra la lista de multimedios.
15. El profesor selecciona un multimedio.	16. El Sistema carga el multimedio, ya sea una imagen, audio o video.
17. El profesor selecciona la opción: usar pizarrón.	18. El Sistema carga el componente del pizarrón para que el usuario pueda dibujar sobre él.
19. El profesor dibuja: líneas, rectángulos, círculos, elipse, texto y subrayar.	20. El Sistema dibuja el objeto seleccionado por el usuario.
21. El profesor selecciona la opción: borrar para limpiar el pizarrón.	22. El Sistema limpia el pizarrón.
23. El alumno selecciona la opción: imprimir.	24. El Sistema manda el área de dibujo a imprimir.
25. El profesor selecciona la opción: usar chat.	26. El Sistema se encarga de cargar el chat.
27. El profesor envía mensajes a todos los participantes del curso.	28. El Sistema envía el mensaje a todos los participantes del curso.
29. El profesor selecciona la opción: compartir escritorio.	30. El Sistema envía la dirección ip del profesor a los demás participantes del curso y carga el escritorio remoto.

31. El profesor selecciona la opción: descargar recursos.	32. El Sistema carga el componente que muestra la lista de los recursos.
33. El profesor selecciona un recurso y la ubicación donde se guardará.	34. El Sistema descarga el recurso desde su ubicación en el servidor y lo guarda en la ubicación del profesor.
35. El profesor cierra el curso.	36. El Sistema descarga el ambiente.

3.1.6.6 Diagrama de secuencia del caso de uso: Previsualizar curso

En la **Figura 3.13** se muestra la interacción con el sistema, cuando el profesor desea previsualizar un curso.

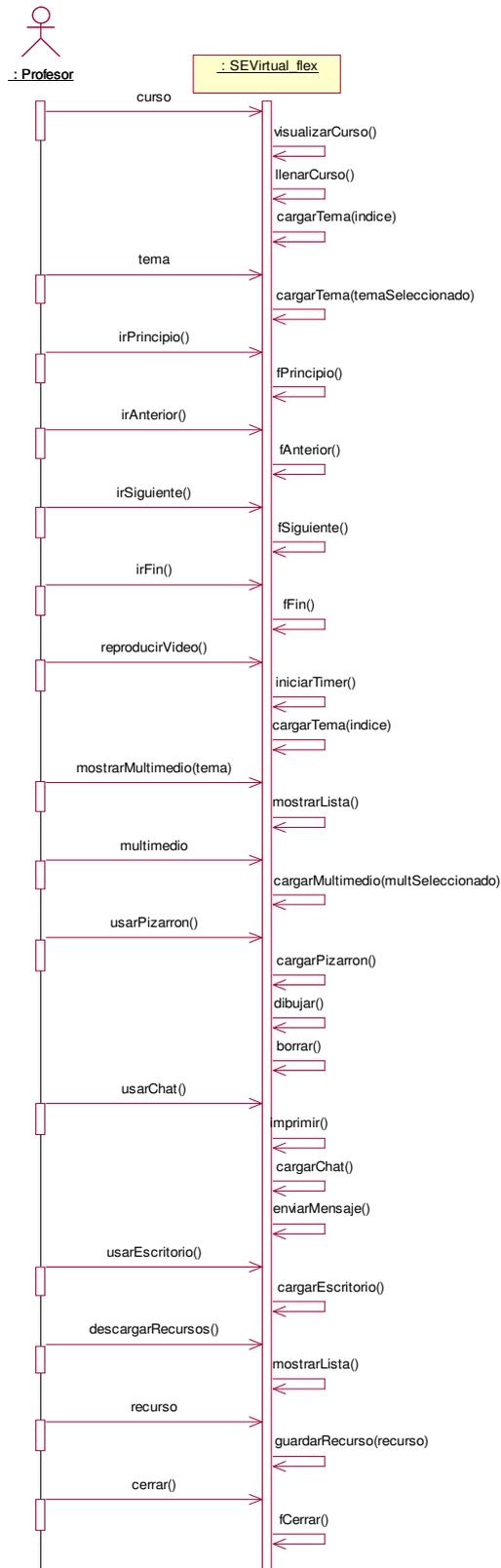


Figura 3.13 Diagrama de secuencia del caso de uso: Previsualizar curso

La **Tabla 3.13** muestra el análisis del caso de uso: ver cursos creados, teniendo como actor al profesor.

Tabla 3.13 Análisis del caso de uso: Ver cursos creados

Caso de uso	
Nombre: Ver cursos creados	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona la opción: Ver cursos creados.	2. El Sistema obtiene todos los cursos registrados por el profesor a través de su correo, una vez obtenidos los cursos genera una estructura de metadatos con el nombre del curso y la ruta de cada curso, por último los nombres de los cursos se cargan en un componente DataGrid.
3. El profesor escoge una de las siguientes opciones: registrar curso, editar curso, previsualizar curso y eliminar curso.	4. El Sistema muestra el ambiente correspondiente a esa opción.

3.1.6.7 Diagrama de secuencia del caso de uso: Ver cursos creados

En la **Figura 3.14** se muestra la interacción con el sistema, cuando el profesor desea ver los cursos que ha creado, para ello el profesor selecciona dicha opción y el sistema se encarga de llenar el componente DataGrid con el nombre de los cursos que ha creado.

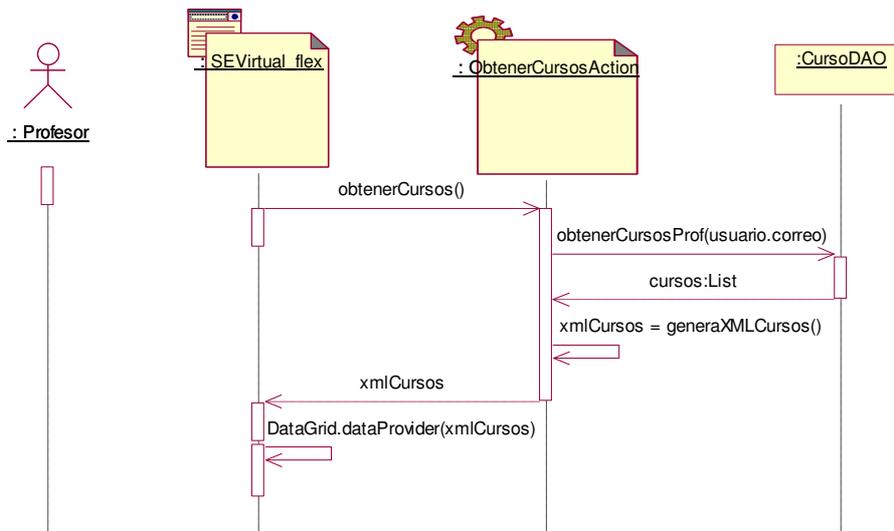


Figura 3.14 Diagrama de secuencia del caso de uso: Ver cursos creados

La **Tabla 3.14** muestra el análisis del caso de uso: registrar tema, teniendo como actor al profesor.

Tabla 3.14 Análisis del caso de uso: Registrar curso

Caso de uso	
Nombre: Registrar curso	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona uno de los cursos creados y escoge la opción: Registrar curso.	2. El Sistema registra el curso a través de su id y muestra un mensaje con el resultado de la operación.
Cursos alternativos	
Línea 2. El Sistema no pudo registrar el curso.	

3.1.6.8 Diagrama de secuencia del caso de uso: Registrar curso

En la **Figura 3.15** se muestra la interacción con el sistema, cuando el profesor desea registrar un curso, para ello el profesor primero selecciona uno de los cursos que ha creado y escoge la opción de registrar, por último el sistema se encarga de registrarlo para que el alumno se pueda inscribir posteriormente.

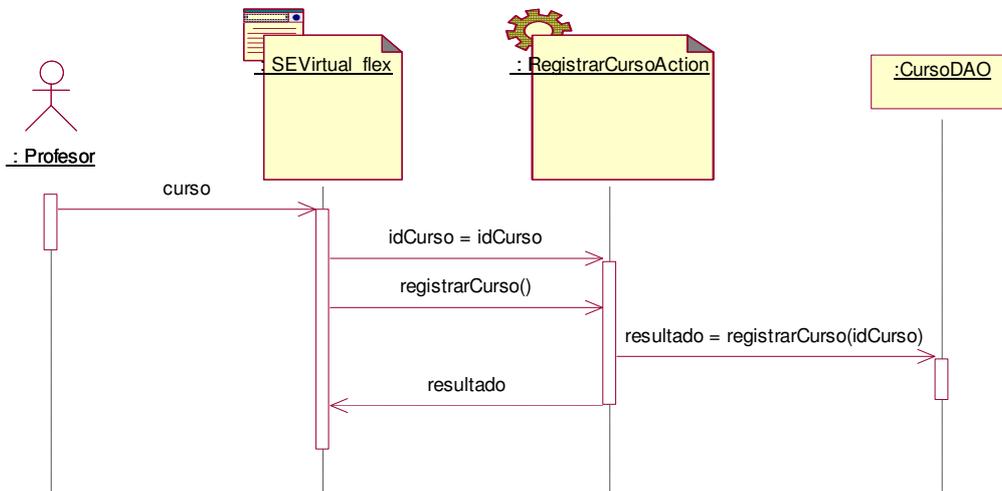


Figura 3.15 Diagrama de secuencia del caso de uso: Registrar curso

La **Tabla 3.15** muestra el análisis del caso de uso: eliminar curso, teniendo como actor al profesor.

Tabla 3.15 Análisis del caso de uso: Eliminar curso

Caso de uso

Nombre: Eliminar curso	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona uno de los cursos creados y escoge la opción: Eliminar curso.	2. El Sistema elimina el curso a través de su id y muestra un mensaje con el resultado de la operación.
Cursos alternativos	
Línea 2. El Sistema no pudo eliminar el curso.	

3.1.6.9 Diagrama de secuencia del caso de uso: Eliminar curso

En la **Figura 3.16** se muestra la interacción con el sistema, cuando el profesor desea eliminar un curso, para ello primero selecciona uno de los curso que ha creado y escoge la opción de eliminar, el sistema se encarga de borrar el curso.

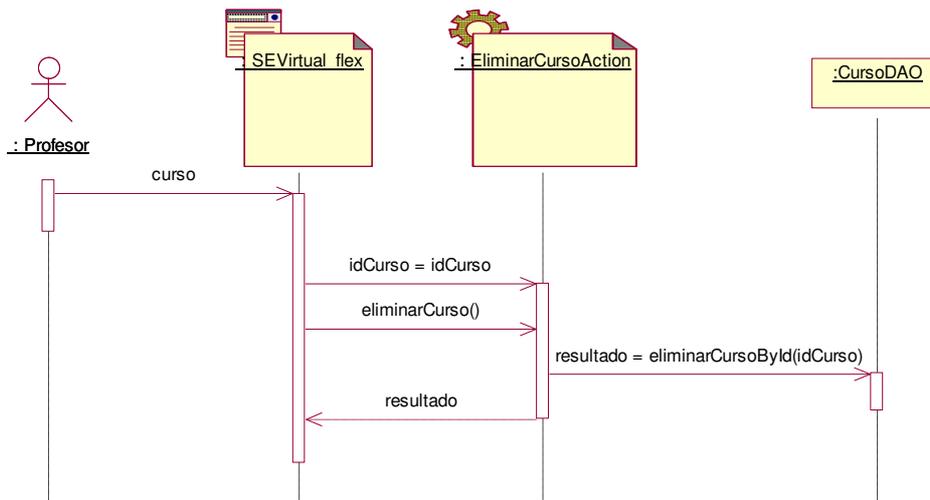


Figura 3.16 Diagrama de secuencia del caso de uso: Eliminar curso

La **Tabla 3.16** muestra el análisis del caso de uso: editar curso, teniendo como actor al profesor.

Tabla 3.16 Análisis del caso de uso: Editar curso

Caso de uso	
Nombre: Editar curso	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema

1. El profesor selecciona uno de los cursos creados y escoge la opción: Editar curso.	2. El Sistema muestra el ambiente de edición de curso.
3. El profesor puede: Agregar un nuevo tema o Modificar un tema.	4. El Sistema muestra el ambiente correspondiente para la opción seleccionada.

3.1.6.10 Diagrama de secuencia del caso de uso: Editar curso

En la **Figura 3.17** se muestra la interacción con el sistema, cuando el profesor desea editar un curso, para ello el profesor debe seleccionar el curso y el sistema se encargará de cargar el ambiente donde podrá editar el curso.

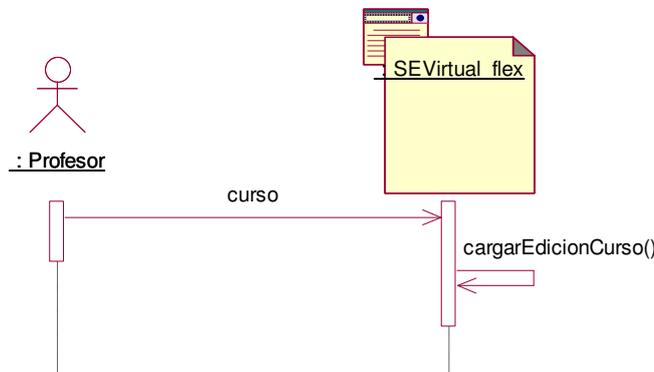


Figura 3.17 Diagrama de secuencia del caso de uso: Editar curso

La **Tabla 3.17** muestra el análisis del caso de uso: modificar tema, teniendo como actor al profesor.

Tabla 3.17 Análisis del caso de uso: Modificar tema

Caso de uso	
Nombre: Modificar tema	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona uno de los temas existentes y selecciona otro componente predefinido.	2. El Sistema muestra el componente a través del cual el profesor podrá insertar el nuevo contenido.

3.1.6.11 Diagrama de secuencia del caso de uso: Modificar tema

En la **Figura 3.18** se muestra la interacción con el sistema, cuando el profesor desea modificar un tema, para ello el profesor primero selecciona un tema para posteriormente poder seleccionar un nuevo componente e insertar el contenido.

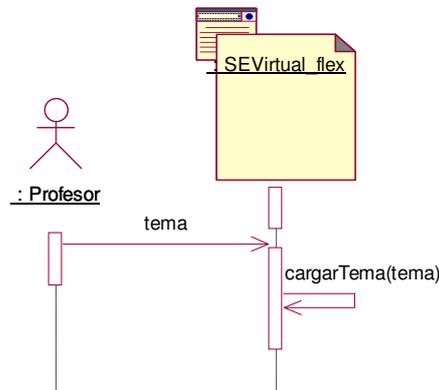


Figura 3.18 Diagrama de secuencia del caso de uso: Modificar tema

La **Tabla 3.18** muestra el análisis del caso de uso: modificar contenido, teniendo como actor al profesor.

Tabla 3.18 Análisis del caso de uso: Modificar contenido

Caso de uso	
Nombre: Modificar contenido	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona uno de los temas existentes.	2. El Sistema muestra el componente y el contenido del tema.
3. El profesor cambia el contenido del curso.	4. El Sistema actualiza el contenido del curso.

3.1.6.12 Diagrama de secuencia del caso de uso: Modificar contenido

En la **Figura 3.19** se muestra la interacción con el sistema, cuando el profesor desea modificar el contenido de un tema, para ello el profesor primero selecciona un tema para posteriormente una vez cargado el contenido modificarlo o insertar un nuevo contenido.

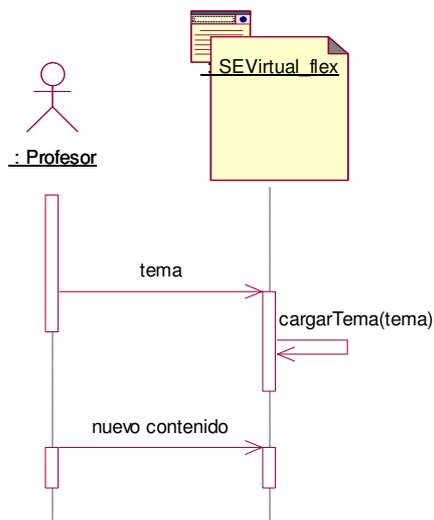


Figura 3.19 Diagrama de secuencia del caso de uso: Modificar contenido

La **Tabla 3.19** muestra el análisis del caso de uso: subir video curso, teniendo como actor al profesor.

Tabla 3.19 Análisis del caso de uso: Subir video curso

Caso de uso	
Nombre: Subir video curso	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona la opción: subir video.	2. El Sistema muestra el componente que permite seleccionar un video .flv.
3. El profesor selecciona el nombre del video.	4. El Sistema envía el video al servidor, crea los directorios y guarda el video con el nombre original.
5. El profesor selecciona el texto explicativo del curso en formato .txt	6. El Sistema envía el archivo de texto al servidor y lo guarda.
Cursos alternativos	
Línea 5. El profesor no quiere guardar ningún texto explicativo del curso.	

La **Tabla 3.20** muestra el análisis del caso de uso: crear pregunta, teniendo como actor al profesor.

Tabla 3.20 Análisis del caso de uso: Crear pregunta

Caso de uso
Nombre: Crear pregunta

Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona la opción: pregunta y puede seleccionar entre los cinco tipos de preguntas que hay: falso verdadero, relación de columnas, opción múltiple, llenado en blanco y selección de área.	2. El Sistema muestra el componente del tipo de pregunta seleccionado.
3. El profesor ingresa los datos de la pregunta.	4. El sistema va creando el meta datos de la pregunta con los datos ingresados.
5. El profesor selecciona un tema y la opción: Guardar pregunta.	6. El Sistema envía el meta datos de la pregunta y lo guarda en el servidor.
Cursos alternativos	
Línea 5. El profesor no selecciona un tema, para ello el sistema le pide que seleccione un tema.	

La **Tabla 3.21** muestra el análisis del caso de uso: crear examen, teniendo como actor al profesor.

Tabla 3.21 Análisis del caso de uso: Crear examen

Caso de uso	
Nombre: Crear examen	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona la opción: examen.	2. El Sistema muestra el componente que permite seleccionar preguntas para el nuevo examen.
3. El profesor selecciona las preguntas que desea agregar al examen.	4. El Sistema crea el meta datos del examen y le va agregando la pregunta que selecciona el profesor.
5. El profesor selecciona la opción: previsualizar	6. El Sistema muestra el componente de evaluación cargando el meta datos generado.
7. El profesor selecciona la opción: guardar examen.	8. El Sistema envía el meta datos del examen y lo guarda en el servidor.
Cursos alternativos	
Línea 5. El profesor no quiere previsualizar el examen se puede ir directamente a la opción: guardar examen.	

La **Tabla 3.22** muestra el análisis del caso de uso: agregar recomendaciones, teniendo como actor al profesor.

Tabla 3.22 Análisis del caso de uso: Agregar recomendaciones

Caso de uso	
Nombre: Agregar recomendaciones	
Actores: Profesor	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El profesor selecciona un tema y la opción: recomendar.	2. El Sistema muestra el componente que permite agregar las recomendaciones del tema, tanto de libros, páginas de Internet y ejercicios.
3. El profesor ingresa los libros y/o páginas de Internet y/o ejercicios.	4. El sistema va creando el meta datos de las recomendaciones.
5. El profesor selecciona la opción: guardar.	6. El Sistema envía agrega al elemento tema del meta datos del curso el meta datos con las recomendaciones.

3.1.7 Diagrama de casos de uso del alumno

La **Figura 3.20** muestra el diagrama de casos de uso del alumno, en el cual se muestran sus actividades principales. Las cuales una vez iniciada su sesión son: ver cursos registrados y ver cursos inscritos. Dentro de la opción ver cursos registrados, el alumno puede obtener un listado de todos los cursos que han registrado los profesores que están dados de alta en el sistema, así como seleccionar uno para inscribirse, una vez inscrito a algún curso puede visualizarlos a través de la opción ver cursos inscritos y si así lo desea puede darse de baja de algún curso al que se halla inscrito. El sistema se encargará de cargar el ambiente donde se visualice el contenido del curso.

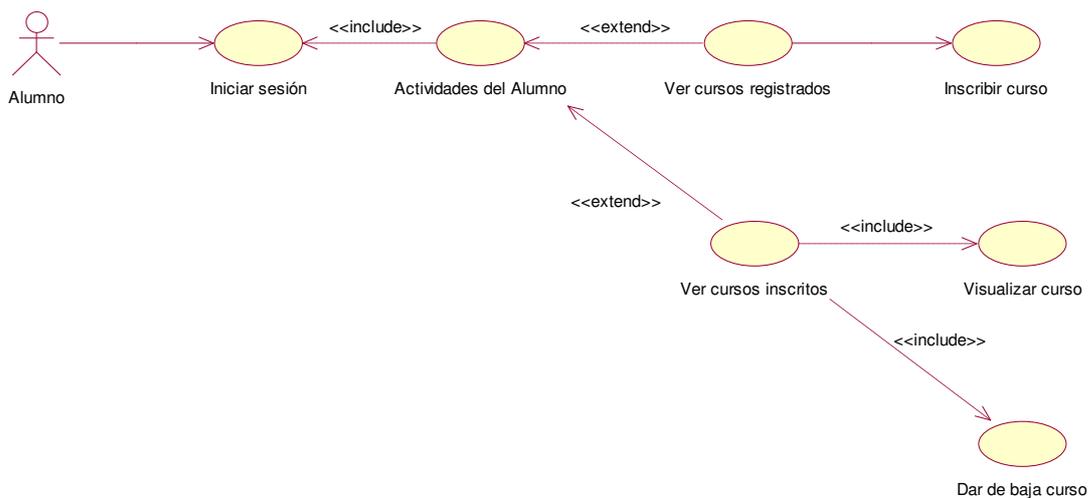


Figura 3.20 Diagrama de casos de uso del alumno

La **Tabla 3.23** muestra el análisis del caso de uso: general del alumno, teniendo como actividades principales: inscribirse a un curso, visualizarlo y darlo de baja.

3.1.8 Análisis del caso de uso del alumno

Tabla 3.23 Análisis del caso de uso del alumno

Caso de uso	
Nombre: Caso de uso del alumno Actores: Alumno	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El alumno inicia sesión.	2. El sistema muestra el menú y la parte del sistema que le corresponde al alumno.
3. El alumno selecciona una de las actividades que desea realizar como son: ver cursos registrados para poder inscribirse a un curso o ver cursos inscritos para poder visualizar alguno o darlo de baja.	4. El Sistema muestra el ambiente correspondiente a esa actividad.

La **Tabla 3.24** muestra el análisis del caso de uso: ver cursos registrados, teniendo como actor al alumno.

Tabla 3.24 Análisis del caso de uso: Ver cursos registrados

Caso de uso	
Nombre: Ver cursos registrados Actores: Alumno	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El alumno selecciona la opción: Ver cursos registrados.	2. El Sistema obtiene todos los cursos que han registrado los profesores y genera la estructura de metadatos con el nombre y el id del curso para cargarla en el componente DataGrid.

3.1.8.1 Diagrama de secuencia del caso de uso: Ver cursos registrados

En la **Figura 3.21** se muestra la interacción con el sistema, cuando el alumno desea ver los cursos registrados por los profesores.

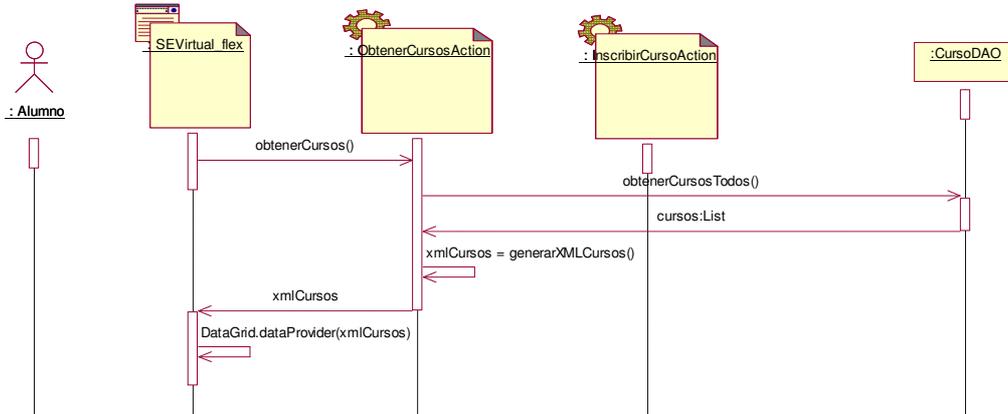


Figura 3.21 Diagrama de secuencia del caso de uso: Ver cursos registrados

La **Tabla 3.25** muestra el análisis del caso de uso: inscribir curso, teniendo como actor al alumno.

Tabla 3.25 Análisis del caso de uso: Inscribir curso

Caso de uso	
Nombre: Inscribir curso	
Actores: Alumno	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El alumno puede inscribirse a un nuevo curso, para ello selecciona el curso al que desea inscribirse.	2. El Sistema guarda el curso al que se inscribió el alumno y muestra un mensaje con el resultado de la operación.

3.1.8.2 Diagrama de secuencia del caso de uso: Inscribir curso

En la **Figura 3.22** se muestra la interacción con el sistema, cuando el alumno desea inscribirse a un curso, para ello el alumno selecciona uno de los cursos y el sistema se encarga de guardar la inscripción al curso.

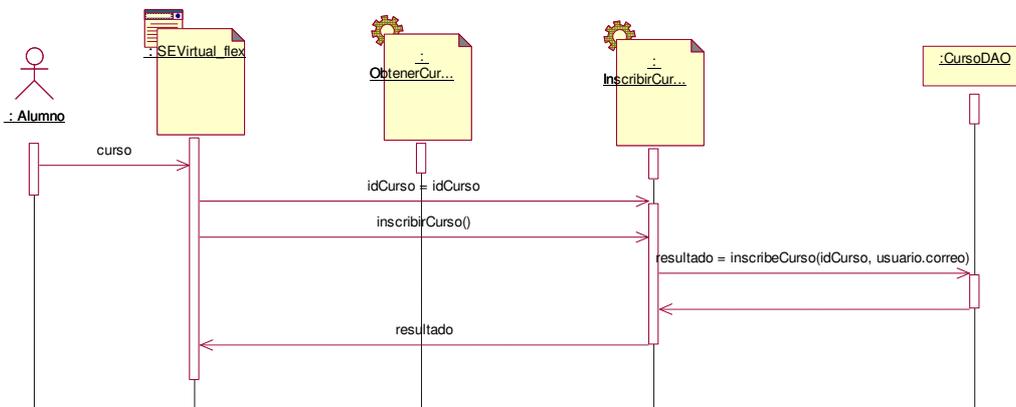


Figura 3.22 Diagrama de secuencia del caso de uso: Inscribir curso

La **Tabla 3.26** muestra el análisis del caso de uso: ver cursos inscritos, teniendo como actor al alumno.

Tabla 3.26 Análisis del caso de uso: Ver cursos inscritos

Caso de uso	
Nombre: Ver cursos inscritos	
Actores: Alumno	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El alumno selecciona la opción: Ver cursos inscritos.	2. El Sistema obtiene todos los cursos a los que se ha inscrito y genera la estructura de metadatos con el nombre y el id del curso para cargarla en el componente DataGrid.

3.1.8.3 Diagrama de secuencia del caso de uso: Ver cursos inscritos

En la **Figura 3.23** se muestra la interacción con el sistema, cuando el alumno desea ver los cursos a los que se ha inscrito, para posteriormente visualizar alguno o darse de baja de un curso.

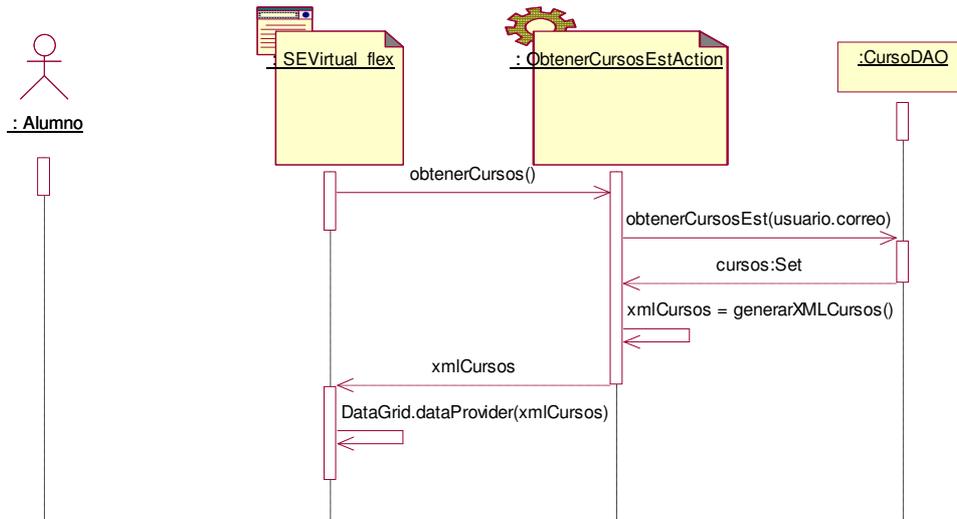


Figura 3.23 Diagrama de secuencia del caso de uso: Ver cursos inscritos

La **Tabla 3.27** muestra el análisis del caso de uso: visualizar curso, teniendo como actor al alumno.

Tabla 3.27 Análisis del caso de uso: Visualizar curso

Caso de uso	
Nombre: Visualizar curso	
Actores: Alumno	

Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
37. El alumno selecciona la opción: visualizar curso.	38. El Sistema muestra el ambiente del curso al que se ha inscrito el alumno, para ello carga la estructura de metadatos del servidor, llena los componentes con la información del curso y carga el tema del índice cero, es decir, el primer tema.
39. El alumno navega por el curso a través del árbol de la estructura del curso o a través de la barra de navegación, el profesor puede seleccionar la opción ir al principio.	40. El Sistema actualiza el índice en la posición cero para cargar el primer tema.
41. El alumno navega por el curso a través de la opción ir al anterior.	42. El Sistema actualiza el índice disminuyéndolo en una unidad y carga el contenido del tema.
43. El alumno navega por el curso a través de la opción ir al siguiente.	44. El Sistema actualiza el índice incrementándolo en una unidad y carga el contenido del tema.
45. El alumno navega por el curso a través de la opción ir al final.	46. El Sistema actualiza el índice en la posición final y carga el contenido del tema.
47. El alumno selecciona la opción: reproducir para comenzar la reproducción del video explicativo del curso.	48. El Sistema comienza la reproducción e inicia el contador para cambiar automáticamente los temas de acuerdo a la reproducción del video.
49. El alumno selecciona un tema y escoge la opción: multimedia para visualizar los multimedia de ese tema.	50. El Sistema carga el componente que muestra la lista de multimedia.
51. El alumno selecciona un multimedia.	52. El Sistema carga el multimedia, ya sea una imagen, audio o video.
53. El alumno selecciona la opción: usar pizarrón.	54. El Sistema carga el componente del pizarrón para que el usuario pueda dibujar sobre él.
55. El alumno dibuja: líneas, rectángulos, círculos, elipse, texto y subrayar.	56. El Sistema dibuja el objeto seleccionado por el usuario.
57. El alumno selecciona la opción: borrar para limpiar el pizarrón.	58. El Sistema limpia el pizarrón.
59. El alumno selecciona la opción: imprimir.	60. El Sistema manda el área de dibujo a imprimir.
61. El alumno selecciona la opción: usar chat.	62. El Sistema se encarga de cargar el chat.
63. El alumno envía mensajes a todos los participantes del curso.	64. El Sistema envía el mensaje a todos los participantes del curso.

65. El alumno selecciona la opción: compartir escritorio.	66. El Sistema envía la dirección ip del alumno a los demás participantes del curso y carga el escritorio remoto.
67. El alumno selecciona la opción: descargar recursos.	68. El Sistema carga el componente que muestra la lista de los recursos.
69. El alumno selecciona un recurso y la ubicación donde se guardará.	70. El Sistema descarga el recurso desde su ubicación en el servidor y lo guarda en la ubicación del alumno.
71. El alumno selecciona la opción: evaluación.	72. El Sistema carga el meta datos del examen en el componente de evaluación iniciando con la primera pregunta.
73. El alumno responde la pregunta y selecciona la opción: siguiente.	74. El sistema evalúa la pregunta y si es incorrecta envía métricas al sistema multiagentes. Al finalizar la evaluación el Sistema le muestra su calificación, la recomendación dinámica junto con las recomendaciones y le reconfigura el curso.
75. El alumno cierra el curso.	76. El Sistema descarga el ambiente.

3.1.8.4 Diagrama de secuencia del caso de uso: Visualizar curso

En la **Figura 3.24** se muestra la interacción con el sistema, cuando el alumno desea visualizar algún curso al que se ha inscrito, para ello primero debe seleccionar un curso y el sistema se encargará de cargar el ambiente del curso.

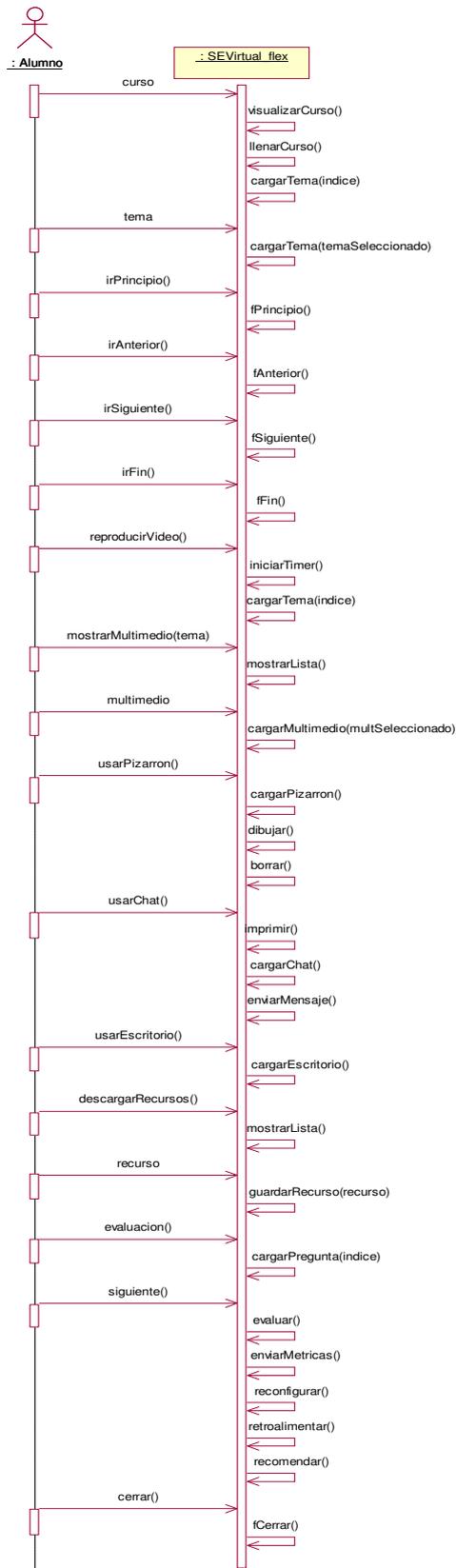


Figura 3.24 Diagrama de secuencia del caso de uso: Visualizar curso

La **Tabla 3.28** muestra el análisis del caso de uso: dar de baja curso, teniendo como actor al alumno.

Tabla 3.28 Análisis del caso de uso: Dar de baja curso

Caso de uso	
Nombre: Dar de baja curso	
Actores: Alumno	
Curso normal de los eventos	
Acción de los actores	Respuesta del Sistema
1. El alumno selecciona uno de los cursos a los que se ha inscrito y escoge la opción: Dar de baja curso.	2. El Sistema elimina el curso de los cursos inscritos a través de su id y muestra un mensaje con el resultado de la operación.
Cursos alternativos	
Línea 2. El Sistema no pudo eliminar el curso de los cursos inscritos para ese alumno.	

3.1.8.5 Diagrama de secuencia del caso de uso: Dar de baja curso

En la **Figura 3.25** se muestra la interacción con el sistema, cuando el alumno desea dar de baja algún curso al que se ha inscrito, para ello primero debe seleccionar un curso y el sistema se encargará de eliminar su inscripción al curso.

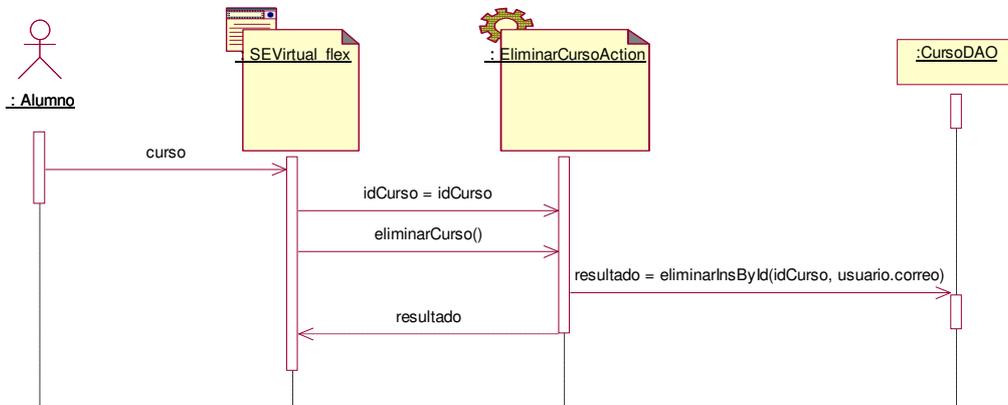


Figura 3.25 Diagrama de secuencia del caso de uso: Dar de baja curso

3.2 DIAGRAMA DE CLASES

3.2.1 Diagrama de clases del servicio: EliminarCursoAction

La **Figura 3.26** muestra el conjunto de clases que interactúan para dar respuesta al servicio: EliminarCursoAction, el cual permite eliminar un curso que haya sido registrado previamente por algún profesor a través de su id. La clase CursoDAO es la que se encarga de eliminar el curso de la base de datos a través del método eliminarCursoById(idCurso), el cual recibe el id del curso.

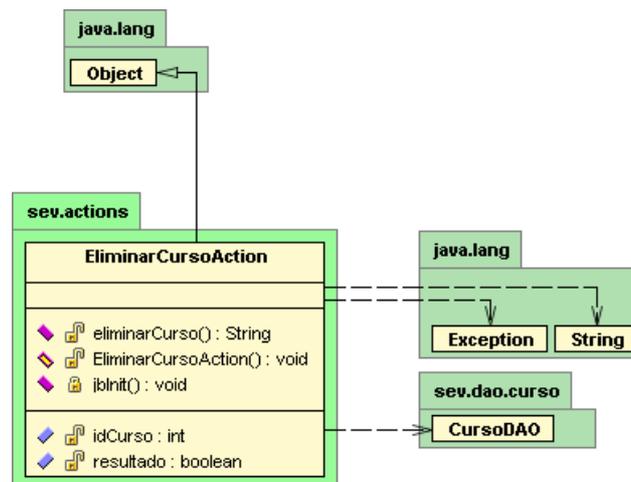


Figura 3.26 Diagrama de clases del servicio: EliminarCursoAction

3.2.2 Diagrama de clases del servicio: EliminarInsCurAction

La **Figura 3.27** muestra el conjunto de clases que interactúan para dar respuesta al servicio: EliminarInsCurAction, el cual permite eliminar la inscripción al curso por parte del alumno a través de su id. La clase CursoDAO a través de su método eliminarInsById(idCurso) elimina la inscripción al curso de la base de datos.

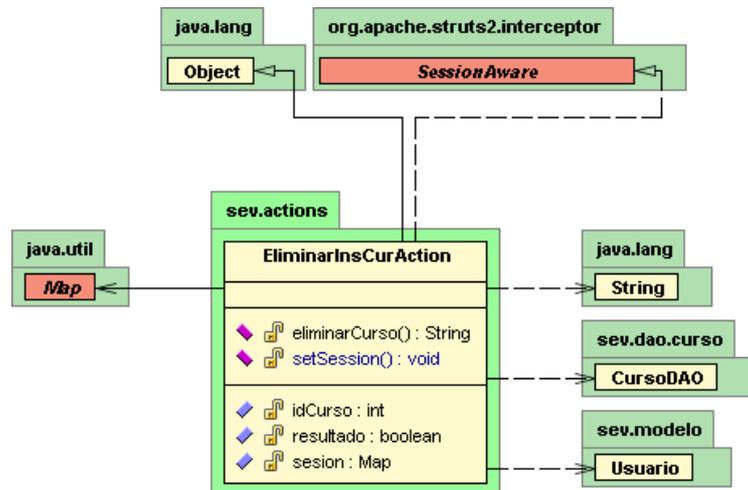


Figura 3.27 Diagrama de clases del servicio: EliminarInsCurAction

3.2.3 Diagrama de clases del servicio: GuardarCursoAction

La **Figura 3.28** muestra el conjunto de clases que interactúan para dar respuesta al servicio: GuardarCursoAction, el cual permite guardar el registro del curso creado por el profesor en la base de datos. La clase EscribirArchivo permite escribir los metadatos del curso en el servidor y la clase CursoDAO permite guardar los datos del curso en la base de datos.

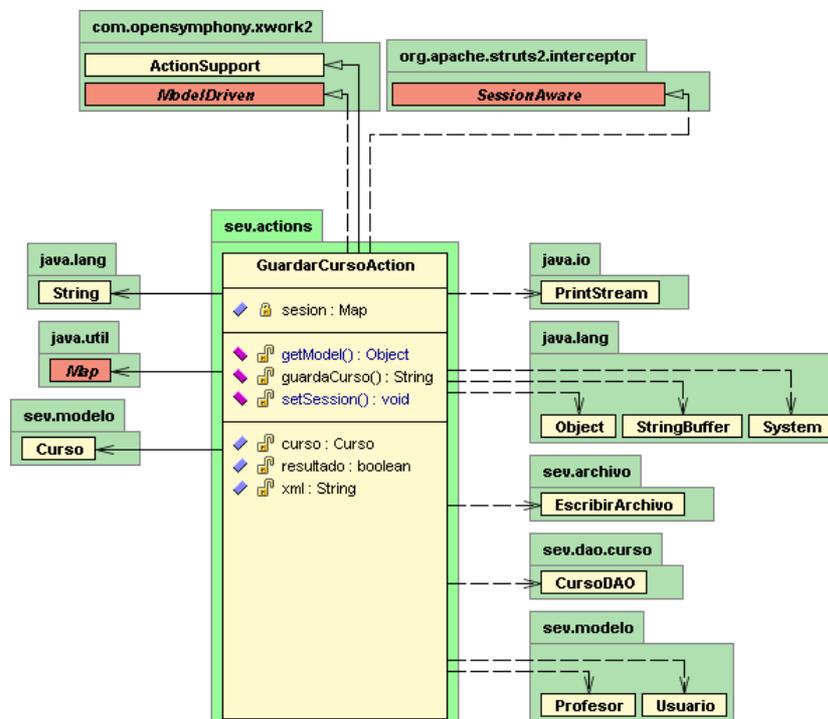


Figura 3.28 Diagrama de clases del servicio: GuardarCursoAction

3.2.4 Diagrama de clases del servicio: GuardarInfoAction

La **Figura 3.29** muestra el conjunto de clases que interactúan para dar respuesta al servicio: GuardarInfoAction, el cual permite guardar la información de los temas que se agregan a los cursos. La clase EscribirArchivo guarda el archivo de metadatos con toda la información del tema en el servidor.

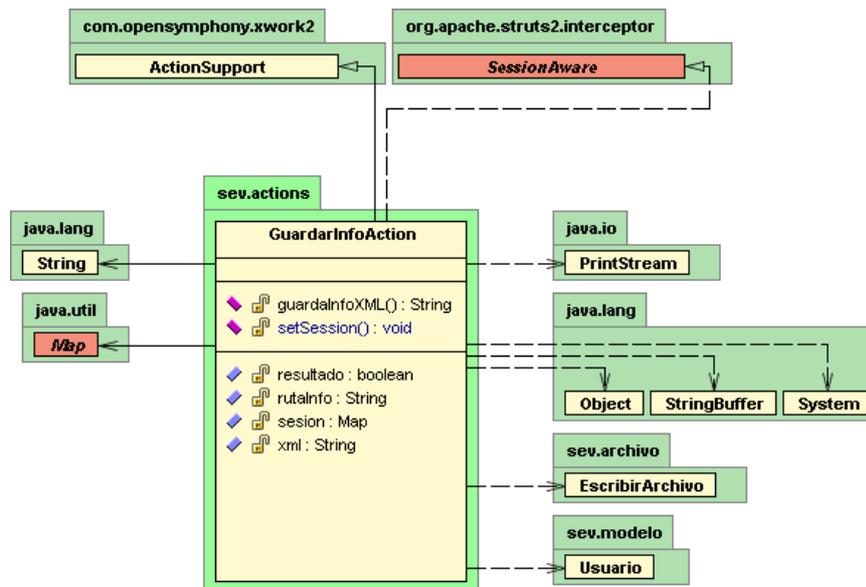


Figura 3.29 Diagrama de clases del servicio: GuardarInfoAction

3.2.5 Diagrama de clases del servicio: GuardarTextoAction

La **Figura 3.30** muestra el conjunto de clases que interactúan para dar respuesta al servicio: GuardarTextoAction, el cual permite guardar el contenido del tema en un archivo de texto dentro del servidor. La clase EscribirArchivo guarda el archivo de texto en el servidor.

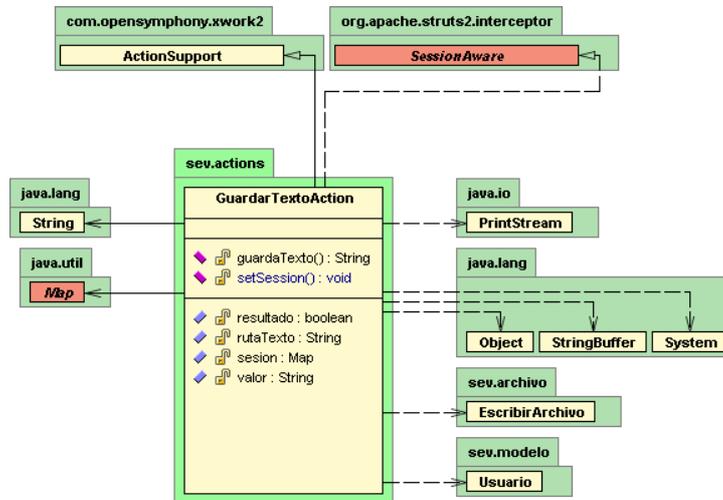


Figura 3.30 Diagrama de clases del servicio: GuardarTextoAction

3.2.6 Diagrama de clases del servicio: InscribirCursoAction

La **Figura 3.31** muestra el conjunto de clases que interactúan para dar respuesta al servicio: InscribirCursoAction, el cual permite al alumno inscribirse a un curso por su id. La clase CursoDAO es la que se comunica con la base de datos para guardar la inscripción del curso.

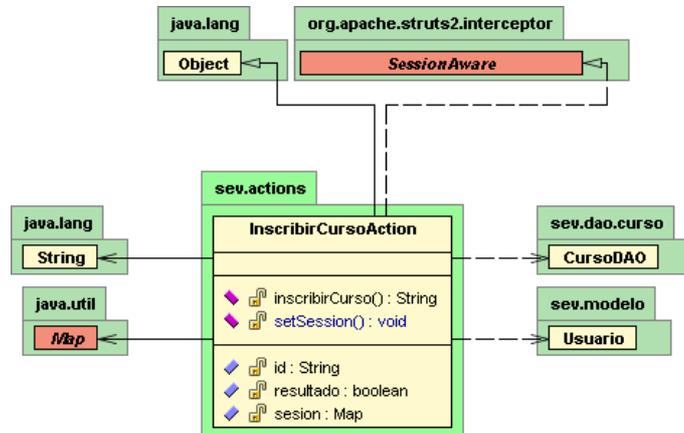


Figura 3.31 Diagrama de clases del servicio: InscribirCursoAction

3.2.7 Diagrama de clases del servicio: LoginAction

La **Figura 3.32** muestra el conjunto de clases que interactúan para dar respuesta al servicio: LoginAction, el cual permite a cualquier usuario (profesor, alumno y administrador) iniciar sesión en el sistema, proporcionando su correo y su *password*. La clase UsuarioDAO se encarga de validar al usuario, verificando que éste exista y que su contraseña sea la correcta.

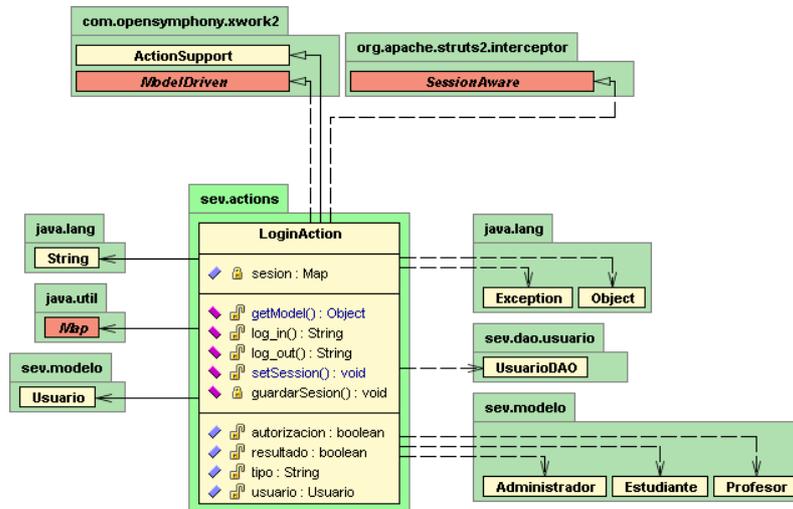


Figura 3.32 Diagrama de clases del servicio: InscribirCursoAction

3.2.8 Diagrama de clases del servicio: ObtenerCursosAction

La **Figura 3.33** muestra el conjunto de clases que interactúan para dar respuesta al servicio: ObtenerCursosAction, el cual se encarga de regresar al cliente una lista con la información de todos los cursos registrados por los profesores, como su id, la ruta y el nombre del curso. La clase CursoDAO se encarga de obtener de la base de datos todos los cursos registrados.

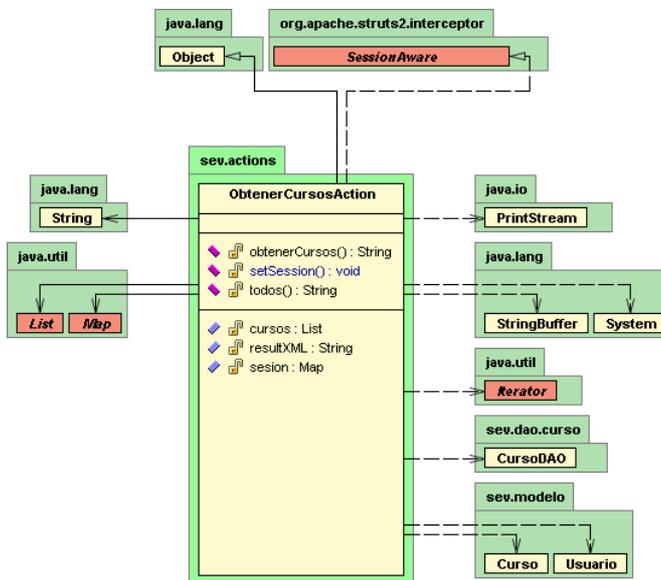


Figura 3.33 Diagrama de clases del servicio: ObtenerCursosAction

3.2.9 Diagrama de clases del servicio: ObtenerCursosEstAction

La **Figura 3.34** muestra el conjunto de clases que interactúan para dar respuesta al servicio: ObtenerCursosEstAction, el cual permite obtener los cursos a los que un alumno se ha inscrito. La clase CursoDAO se encarga de obtener todos los cursos de la base de datos.

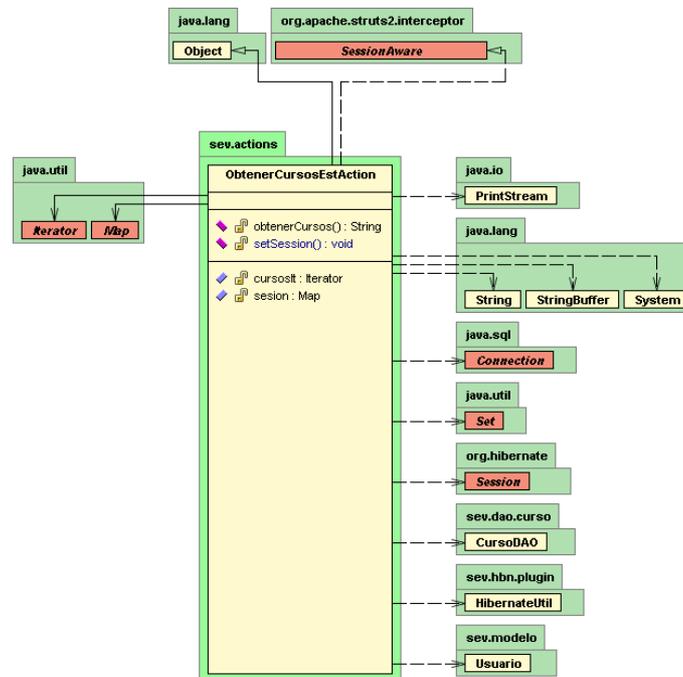


Figura 3.34 Diagrama de clases del servicio: ObtenerCursosAction

3.2.10 Diagrama de clases del servicio: RegistrarUsuarioAction

La **Figura 3.35** muestra el conjunto de clases que interactúan para dar respuesta al servicio: RegistrarUsuarioAction, el cual permite que nuevos usuarios se registren dentro del sistema. La clase UsuarioDAO registra todos los datos del usuario como: el nombre y el correo, dentro de la base de datos.

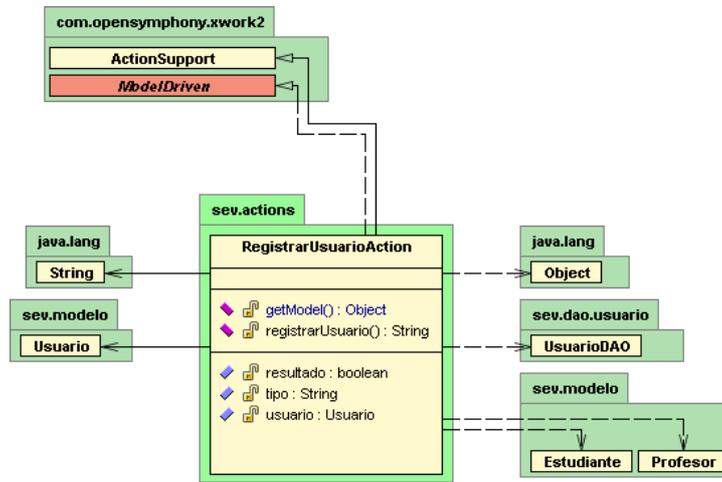


Figura 3.35 Diagrama de clases del servicio: RegistrarUsuarioAction

3.2.11 Diagrama de clases del servicio: UploadAction

La **Figura 3.36** muestra el conjunto de clases que interactúan para dar respuesta al servicio: UploadAction, el cual permite subir el flujo de datos del archivo multimedia al servidor, los tipos de archivos permitidos son los: .gif, .jpg, .png, .flv y .mp3.

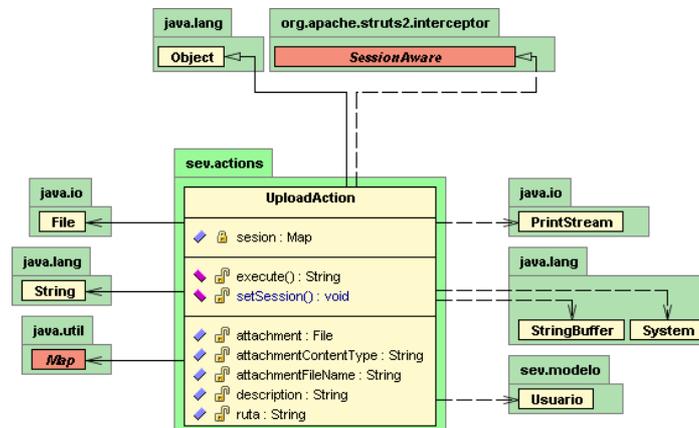


Figura 3.36 Diagrama de clases del servicio: UploadAction

3.2.12 Diagrama de clases del servicio: UsuarioAction

La **Figura 3.37** muestra el conjunto de clases que interactúan para dar respuesta al servicio: UsuarioAction, el cual se encarga de varias operaciones como: buscar, actualizar y eliminar un usuario de la base de datos. La clase UsuarioDAO busca, elimina y actualiza al usuario a través de su id.

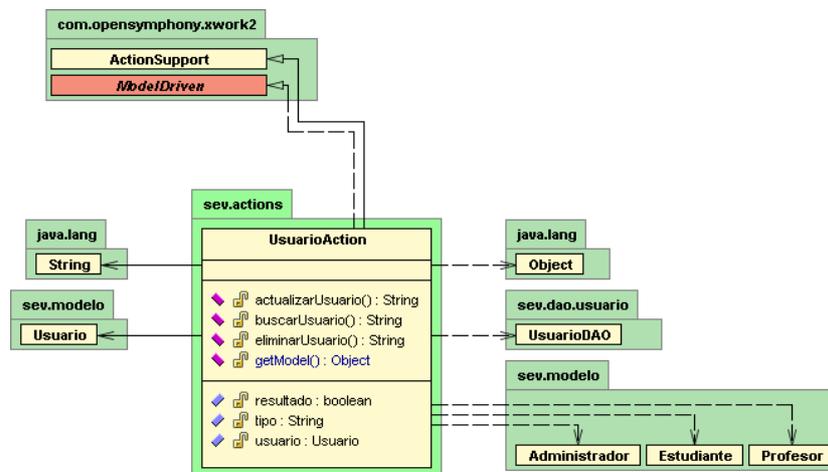


Figura 3.37 Diagrama de clases del servicio: UsuarioAction

3.2.13 Diagrama de clases de la clase: CursoDAO

La **Figura 3.38** muestra la clase que interactúa con la base de datos para dar respuesta a las operaciones que se pueden realizar con un curso y que están disponibles para algunos de los servicios descritos anteriormente, es decir, las operaciones básicas de altas, bajas y cambios tanto para los cursos como para las inscripciones.

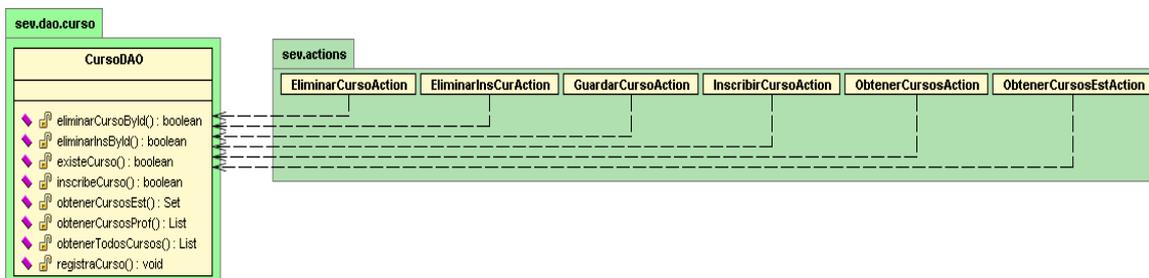


Figura 3.38 Diagrama de clases de la clase: CursoDAO

3.2.14 Diagrama de clases de la clase: UsuarioDAO

La **Figura 3.39** muestra la clase que interactúa con la base de datos para dar respuesta a las operaciones de altas, bajas y cambios de un usuario, disponibles para algunos de los servicios descritos anteriormente.

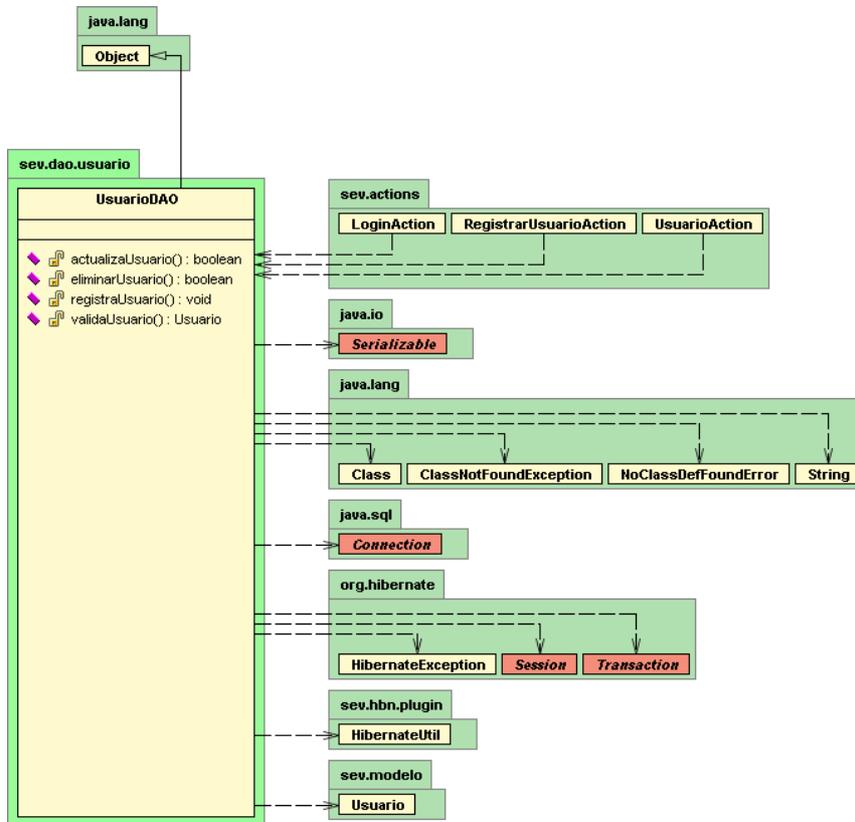


Figura 3.39 Diagrama de clases de la clase: UsuarioDAO

3.3 DIAGRAMA DE PAQUETES

3.3.1 Diagrama de paquetes del middleware del Servidor

La **Figura 3.40** muestra el diagrama de paquetes para el conjunto de clases que dan respuesta a los servicios descritos en los diagramas de clases anteriores, este conjunto de clases se encuentra en el middleware del Servidor, el cual se encarga de dar respuesta a todos los servicios que requiere la aplicación del Cliente.

El paquete: `sev.actions` contiene a todas las clases de los servicios; el paquete: `sev.archivo` contiene la clase: `EscribirArchivo`, la cual se encarga de guardar el archivo que se le solicite en el servidor; el paquete: `sev.utils` contiene constantes utilizadas con frecuencia por los actions; el paquete: `sev.dao.curso` contiene la clase `CursoDAO`; el paquete: `sev.dao.usuario` contiene la clase `UsuarioDAO`; y finalmente el paquete: `sev.modelo` contiene todas las clases entidad como son: `Usuario`, `Profesor`, `Estudiante`, `Administrador` y `Curso`.

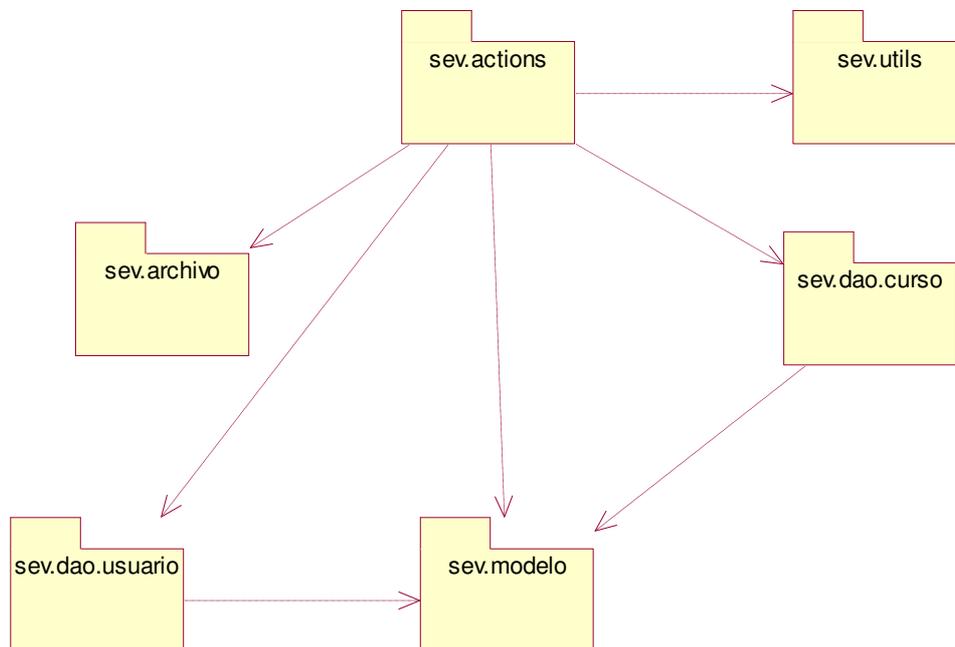


Figura 3.40 Diagrama de paquetes del middleware del Servidor

3.3.2 Diagrama de paquetes de la aplicación del Cliente

La **Figura 3.41** muestra el diagrama de paquetes para el de clases y componentes que se utilizan en la aplicación del cliente para desplegar tanto los menús de los diferentes usuarios, como los escenarios de las opciones que pueden realizar dentro de la aplicación.

Dentro del paquete `curso` se encuentran los componentes y la interfaz que muestra la visualización del curso; en el paquete `estudiante` se encuentran los componentes y la interfaz que muestra el menú del estudiante y las opciones que le corresponden como son: visualizar curso, inscribir curso y ver cursos; el paquete `profesor` contiene los componentes y la interfaz para mostrar el menú del profesor y permitirle realizar sus respectivas opciones; el paquete `clases` contiene clases importantes como: `Curso`, `Tema`, `Multimedia` y aquellas relacionadas con componentes que despliegan el tema; el paquete `util` contiene la clase `Constantes` donde se guardan aquellas constantes importantes; el paquete `contenido` contiene los componentes que despliegan los temas para la parte de edición del curso; el paquete `contenidoVista` contiene los componentes que despliegan los temas para la parte de visualización del curso; el paquete `login` contiene la interfaz para iniciar sesión en el sistema; el paquete `registro` contiene el formulario para el registro de los nuevos usuarios, tanto profesores como alumnos; y finalmente el paquete `marco` contiene el componente `Encabezado` que es común a todas las interfaces del sistema.

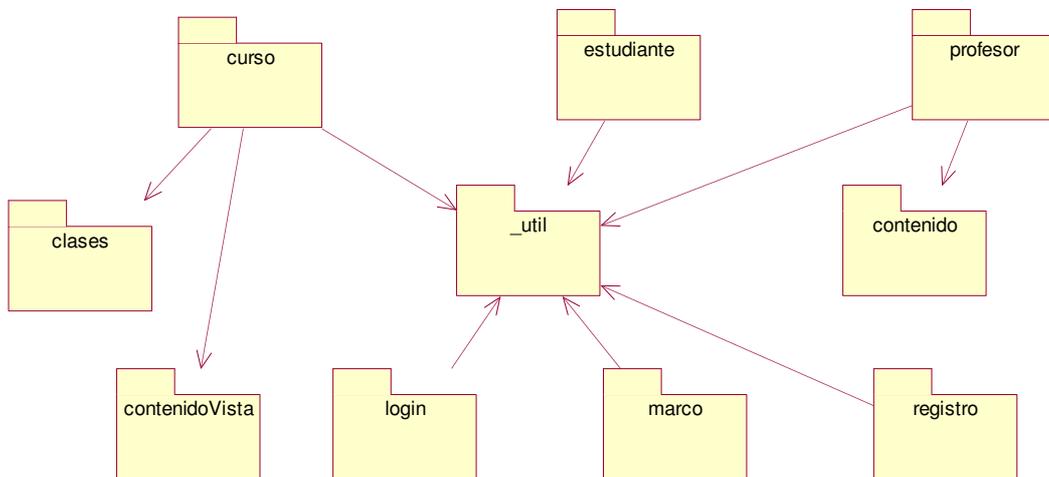


Figura 3.41 Diagrama de paquetes de la aplicación del Cliente

3.4 DIAGRAMA DE COMPONENTES

3.4.1 Diagrama de componentes del paquete: curso

La **Figura 3.42** muestra el diagrama de componentes para el paquete `curso`, donde la interfaz: `VerCurso`, interactúa con los componentes: `Indice`, `Pizarron`, `AreaContenido`, `AreaExplicación`, `BarraNavegación`, `Participantes`, `Chat` y `Escritorio` para mostrar el contenido del curso y las opciones para llevar a cabo la comunicación entre los participantes del curso. El componente `AreaExplicación` despliega un árbol con el índice del curso y el componente `AreaContenido` despliega los componentes asignados a cada tema, así como el contenido del tema.

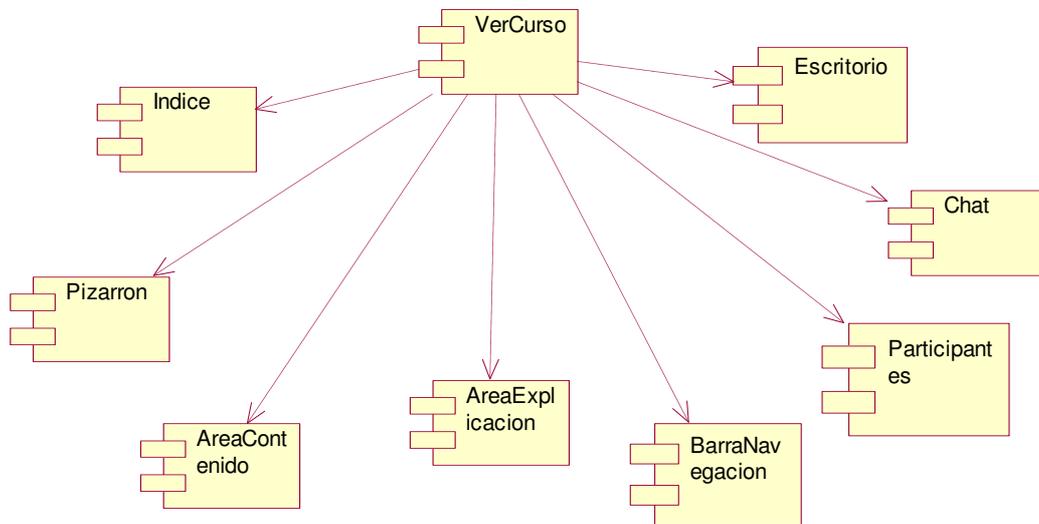


Figura 3.42 Diagrama de componentes para el paquete: curso

3.4.2 Diagrama de componentes del paquete: profesor

La **Figura 3.43** muestra el diagrama de componentes para el paquete profesor, donde la interfaz: MenuProfesor, interactúa con los componentes: VerCursos, NuevoCurso y EditarTema para permitirle al profesor agregar nuevos cursos, ver los cursos que ha creado y editarlos.

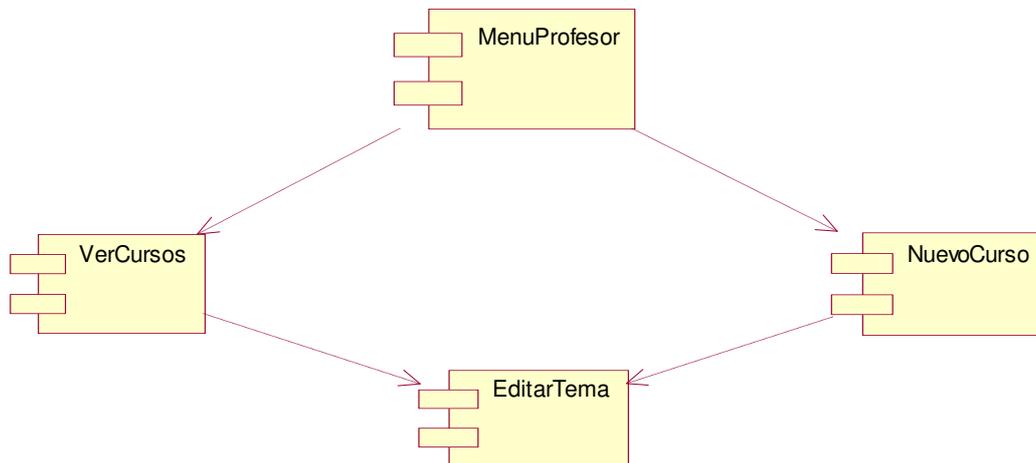


Figura 3.43 Diagrama de componentes para el paquete: profesor

3.4.3 Diagrama de componentes del paquete: alumno

La **Figura 3.44** muestra el diagrama de componentes para el paquete alumno, donde la interfaz: MenuEstudiante, interactúa con los componentes: MisCursos e InsCurso para permitirle al estudiante ver los cursos a los que se ha inscrito e inscribirse a los cursos que ofrecen los profesores.

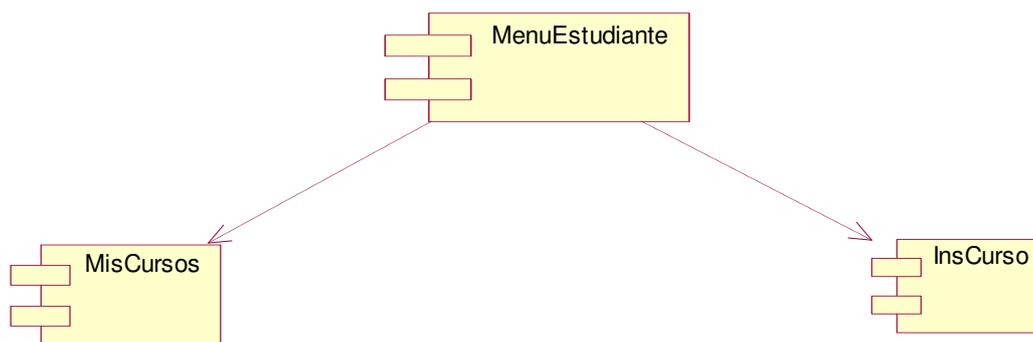


Figura 3.44 Diagrama de componentes para el paquete: alumno

3.5 ESTRUCTURA DEL CURSO

La organización del curso de entrenamiento está basada en el estándar internacional **ADL**, para ello se utiliza una estructura de metadatos, lo cual nos permite separar la parte del contenido del curso de la navegación.

La **Figura 3.45** muestra la estructura de metadatos para la organización del curso, donde los temas se encuentran agrupados por el elemento curso el cual tiene como atributo el nombre del curso, los temas pueden contener subtemas y así sucesivamente, éstos se encuentran anidados dentro de dicho elemento y multimedia, ambos elementos tienen los mismos atributos: tipo, ruta y nombre.

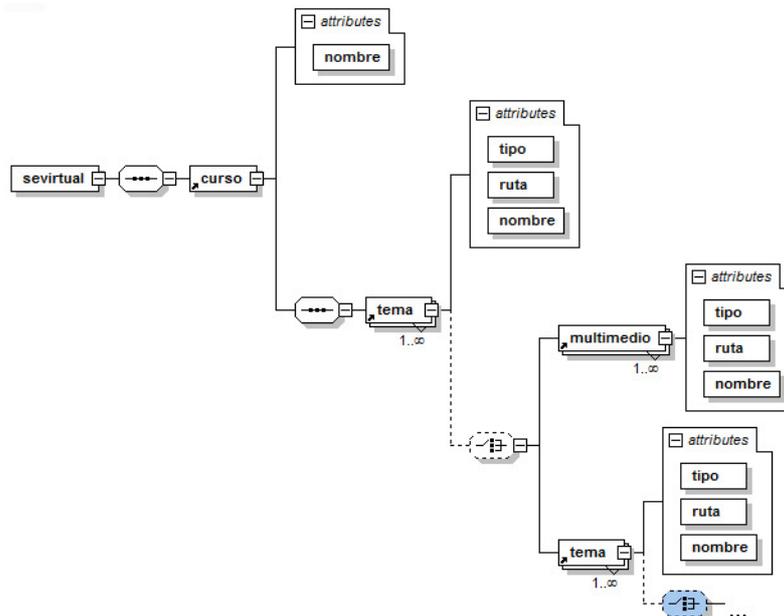


Figura 3.45 Estructura de metadatos para la organización del curso

Para cada tema se crea una estructura de metadatos correspondiente con el tipo de componente que se eligió para dicho tema, el tipo de componente puede ser: solo texto, título doble texto, solo título, título subtítulo y título texto.

La **Figura 3.46** muestra la estructura de metadatos asociada al tema con la información del tipo de componente seleccionado, en este ejemplo se muestra para tipo de componente: título doble texto, en el cual se guarda la información del título y de las rutas donde se encuentran los archivos de texto que almacenan el contenido del tema.

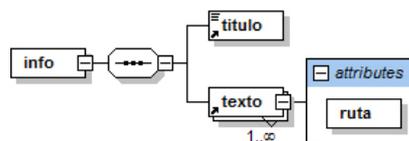


Figura 3.46 Estructura de metadatos para el tipo de componente: título doble texto

3.6 RESUMEN

En el presente capítulo se abordó el análisis y diseño de SEVirtual, utilizando UML como el lenguaje para modelar estas etapas, además se mostraron los diagramas de casos de uso y los diagramas de secuencia principales. Los diagramas de clases muestran las relaciones entre las clases y el diagrama de paquetes muestra la comunicación entre los paquetes, además se integraron los diagramas de componentes y la estructura de metadatos para la organización del curso de entrenamiento.

En el capítulo siguiente se presenta la implementación de SEVirtual, con los detalles principales de la construcción del sistema, entre ellos las tecnologías y patrones utilizados, así como las clases y componentes más importantes.

CAPÍTULO 4

IMPLEMENTACIÓN DE SEVIRTUAL

En el presente capítulo se aborda la implementación de SEVirtual, se han utilizado patrones de diseño en el sistema, que son formas de encapsular el conocimiento con una estrategia para resolver problemas frecuentes en un contexto específico, los patrones definen una relación entre el problema y la solución en el dominio. El patrón de diseño y la arquitectura que se han utilizado en el sistema son: patrón de composición y la arquitectura **Modelo Vista Controlador** (*Model View Controller*, MVC por sus siglas en inglés), los cuales han sido implementados a través de Adobe Flex 3 [54] para la aplicación *Web* del cliente y Struts 2 [55] para el *middleware* del servidor respectivamente. Otra arquitectura importante utilizada es la arquitectura del **Sistema Multi-Agente** (*Multi-Agent System*, MAS por sus siglas en inglés), la cual es implementada haciendo uso del modelo de **Creencias-Deseos-Intenciones** (*Belief-Desire-Intention*, BDI por sus siglas en inglés), para los estados mentales de los agentes e implementados a través del *framework* de código abierto JADEX [56].

Además, se describen las clases y componentes principales tanto para la aplicación del cliente como para el *middleware* del servidor, así como el uso de tecnologías libres que se han utilizado para el desarrollo de la aplicación, la estructura de los archivos de meta datos y por último se muestra el diagrama relacional de la base de datos utilizada.

4.1 PATRÓN DE COMPOSICIÓN

El patrón de composición permite crear sistemas complejos compuestos por componentes más pequeños, haciendo uso de una interfaz simplificada.

A continuación se listan las ventajas principales del patrón de composición.

4.1.1 Ventajas del patrón de composición

La ventaja principal del patrón de composición es el manejo de los componentes indivisibles y compuestos de la misma manera a través de una interfaz común. Los componentes que conforman al sistema pueden estar constituidos de componentes individuales o contenedores que representan colecciones de componentes.

La **Figura 4.1** muestra un diagrama de clases del patrón de composición del sistema, en donde un componente puede ser un componente indivisible o un componente compuesto; el componente generación de contenido contiene al componente tema y multimedia para agregar multimedios a cada tema. El componente tema puede contener cualquiera de los siguientes componentes: Texto, Título doble texto, Título, Título subtítulo y Título texto; dependiendo de la selección del profesor, en cada componente se puede editar el contenido de los temas, los métodos principales de un componente son: agregar y eliminar un componente, obtener un hijo y la operación que es el

método que describe el funcionamiento del componente. Con los métodos agregar y eliminar componente se puede llevar a cabo la reutilización a nivel de componentes. Los modelos de composición para los demás componentes son similares.

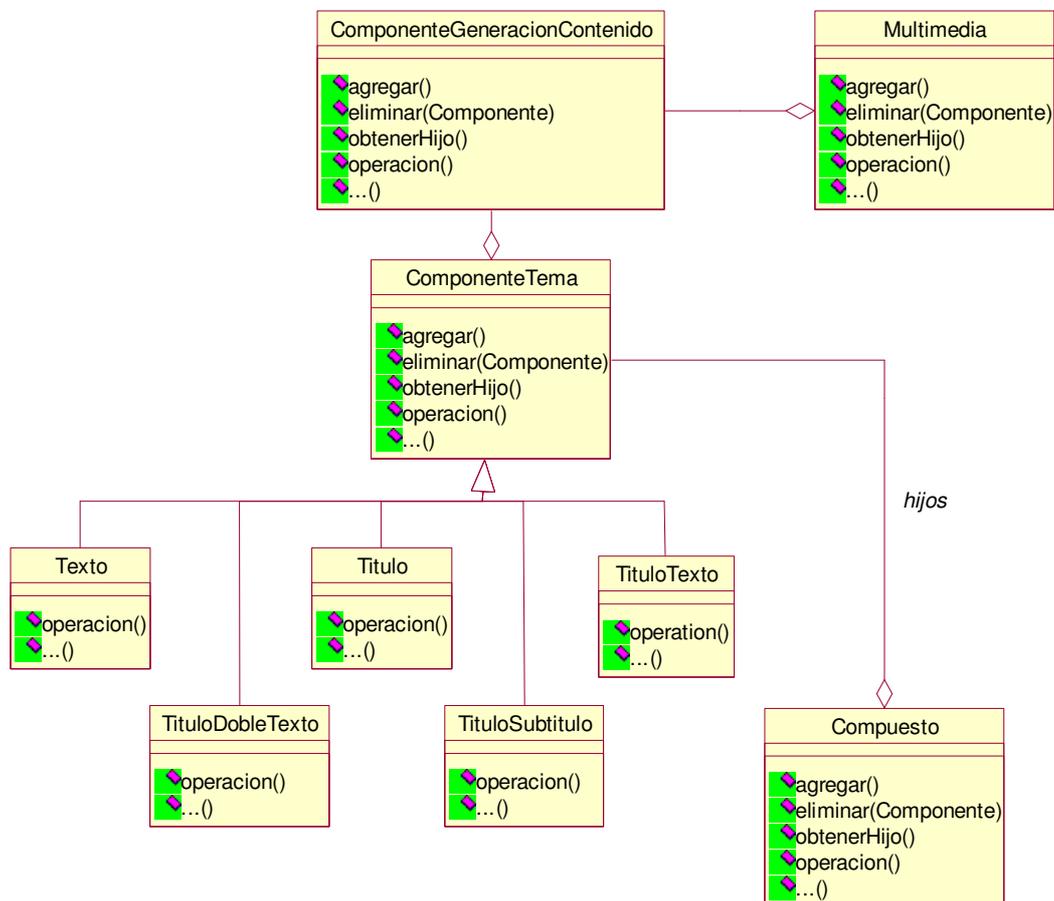


Figura 4.1 Diagrama de clases del patrón de composición

El uso de meta datos para la configuración de los componentes en tiempo de ejecución permite separar la parte del contenido de la navegación de los cursos, además de maximizar la reusabilidad de los componentes al cargar y descargar los meta datos para el mismo componente.

Los componentes también están preparados para el cambio, es decir, si se actualiza la funcionalidad de algún componente o se agregan nuevos componentes, los componentes del sistema no se ven afectados, siempre y cuando se respete la interfaz.

El uso de componentes tiene otras ventajas, por un lado al reducir la complejidad en cuanto a aspectos técnicos en la elaboración de los cursos de entrenamiento y por otro al dividir el problema en problemas más pequeños que se puedan resolver mediante los componentes.

Para la implementación del patrón de composición dentro de SEVirtual se desarrollaron algunos componentes dentro de la aplicación del cliente haciendo uso de una **Aplicación de Internet Rica** (*Rich Internet Application*, RIA por sus siglas en inglés) [57].

Una aplicación RIA es una experiencia atractiva, interactiva, ligera y flexible. Sus características principales son las siguientes:

- Comunicación bidireccional con el servidor, sin recargar la página.
- Integración de multimedia.
- Integración de reproductores multimedia.
- Aplicaciones interactivas.
- Mejora el rendimiento al no tener que recargar toda la página.

Para el desarrollo y despliegue de la aplicación RIA se utilizó Adobe Flex 3, un *framework* de código abierto, el cual se ha establecido como la principal plataforma para la creación de RIAs.

4.2 ARQUITECTURA MODELO VISTA CONTROLADOR

La arquitectura Modelo Vista Controlador (MVC) es una arquitectura para crear aplicaciones complejas. Consta de tres elementos:

- **Modelo:** Contiene los datos y la lógica de la aplicación.
- **Vista:** Presenta la interfaz del usuario y el estado de la aplicación en la pantalla.
- **Controlador:** Maneja la entrada para cambiar el estado de la aplicación.

Las ventajas principales de esta arquitectura son:

- Reusabilidad
- Mantenibilidad

La **Figura 4.2** muestra la implementación de la arquitectura MVC en el sistema, es decir, cómo interactúa el modelo, la vista y el controlador a través de la arquitectura Cliente/Servidor. Del lado del cliente se tiene la aplicación RIA y del lado del servidor el *middleware* que implementa la arquitectura **MVC**, el cliente hace peticiones al servidor, y el servidor envía la respuesta en formato XML. La información de los cursos, las preguntas y los exámenes que se guardan del lado del servidor utilizan una estructura de meta datos de la cual se hablará más a detalle en la sección 4.6 y 4.7, así como las tecnologías de software libre utilizadas, entre ellas, BlazeDS [58], UltraVNC [59], Hibernate [60], entre otras.

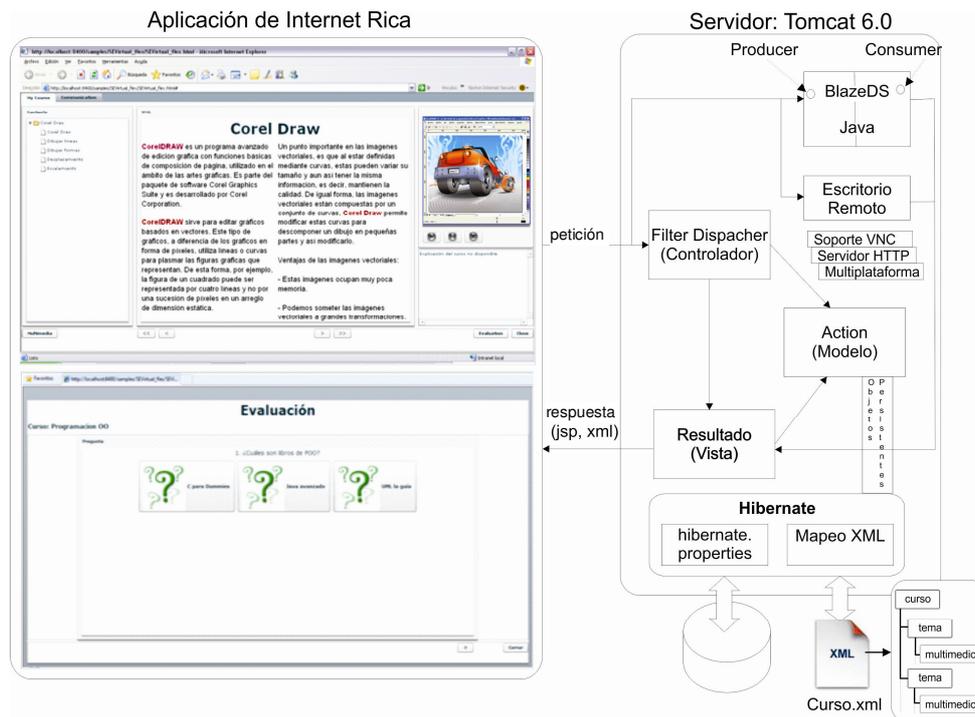


Figura 4.2 Arquitectura MVC y Cliente/Servidor

Para desarrollar dicha arquitectura se utilizó Struts 2 para el *middleware* del servidor, el cual implementa el controlador a través del servlet *Filter Dispatcher*, el modelo a través de acciones o clases desarrolladas en Struts 2 y las vistas se presentan a través de JSP's.

4.3 ARQUITECTURA MULTIAGENTES

La arquitectura del Sistema Multi-Agente (MAS) se basa en la arquitectura IEEE 1484 – LTSA y en la implementación del modelo de Creencias-Deseos-Intenciones (BDI) a través del *framework* de código abierto JADEX.

El modelo BDI [62] fue propuesto por Bratman y es una arquitectura que permite crear agentes de software racionales, dichos agentes a través de ciertas creencias o base de creencias, tratan de alcanzar metas a través de la ejecución de ciertos planes.

JADEX es un *framework* que utiliza Java para la creación de agentes de software proactivos que orientan su comportamiento al logro de metas, implementando el modelo BDI en el sistema. En Jadex las creencias son cualquier tipo de objeto Java almacenado en la base de creencias (*beliefbase*), las metas representan motivaciones concretas (ejemplo: estados que deben de alcanzarse) que influyen en el comportamiento del agente y los planes son clases en Java que utiliza el agente para alcanzar sus metas.

También se ha utilizado un puente para comunicar JADEX con la *Web* en el sistema, que se denomina JADEX-WebBridge, este puente permite comunicar la plataforma JADEX con la aplicación RIA del sistema.

El MAS implementa un tutor virtual, el cual una vez que el estudiante es evaluado, le brindará retroalimentación dinámica, reconfiguración del curso en base a los temas que se le dificulten y recomendaciones en cuanto a libros, ejercicios y páginas de Internet de dichos temas, todo esto a través del logro de metas por parte del tutor virtual. La **Figura 4.3** muestra la implementación del tutor virtual a través del MAS, utilizando las tecnologías de Struts2, Hibernate, JADEX y JADEXWebBridge. Durante la evaluación del estudiante se envían ciertas métricas a través de los componentes al Agente Coordinador, dicho agente delega la responsabilidad al Agente Coach, éste se encarga de guardar dichas métricas en la base de datos para su uso posterior, el Agente Coach una vez terminada la evaluación utilizará dichas métricas para dar retroalimentación dinámica al estudiante, reconfigurarle el curso de acuerdo a los temas que se le hayan dificultado y darle recomendaciones en base a libros, páginas de internet y ejercicios. Las métricas que se envían al MAS son en base a la evaluación del estudiante, como calificaciones, aciertos, errores, temas, tiempos en cada pregunta, etc.

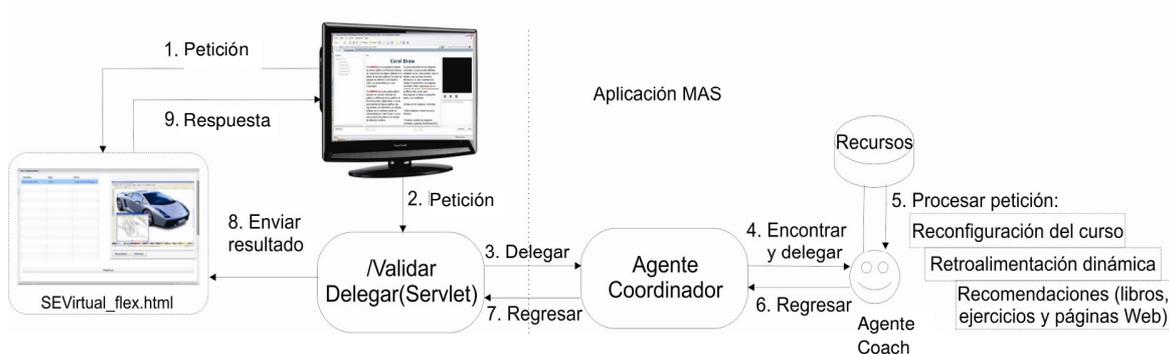


Figura 4.3 Tutor virtual con el MAS

A continuación se presentan los módulos principales de SEVirtual.

4.4 MÓDULOS DE SEVIRTUAL

SEVirtual se divide en cinco módulos principales con sus respectivos sub-módulos de acuerdo a los casos de uso principales identificados en el capítulo anterior.

- Módulo de Login
- Módulo del administrador
 - Alta de usuario
 - Borrar usuario
 - Modificar usuario
- Módulo del profesor
 - Crear curso
 - Ver cursos creados

- Registrar curso
- Eliminar curso
- Previsualizar curso
- Editar curso
- Crear pregunta
- Crear evaluación

- Módulo del alumno
 - Ver cursos registrados
 - Inscribir curso
 - Ver cursos inscritos
 - Visualizar curso
 - Dar de baja curso
 - Ser evaluado

- Módulo del MAS
 - Guardar métrica
 - Reconfigurar curso
 - Retroalimentar dinámicamente
 - Recomendar

Para la creación de los cursos, preguntas y evaluaciones se utilizan estructuras de meta datos para guardar la información y para la administración de los usuarios, preguntas, métricas y cursos, se utiliza una base de datos relacional implementada en el manejador de bases de datos MySQL. La estructura de los meta datos y el modelo relacional de la base de datos se explicarán en la sección 4.6, 4.7 y 4.8. Para la implementación de la comunicación con la base de datos del lado del servidor se utilizó el *framework* Hibernate, un marco de trabajo para el Mapeo Objeto-Relacional (*Object Relational Mapping*, ORM por sus siglas en inglés), que mapea los datos almacenados en la base de datos a objetos persistentes en Java.

A continuación se describen las clases y los componentes principales utilizados en cada módulo.

4.4.1 Módulo de Login

Los componentes y clases utilizados para este módulo se muestran tanto para la aplicación RIA del cliente como para el *middleware* del servidor.

4.4.1.1 RIA del cliente

La **Tabla 4.1** describe el componente principal para el módulo de Login.

Componente: Login.mxml	
Descripción:	Presenta el formulario que permite introducir los campos de correo y contraseña para el inicio de sesión en el sistema.
Atributos:	-
Método:	validarEntrada()
Parámetros que recibe:	ResultEvent
Regresa:	-
Descripción:	Este método se manda a llamar una vez que el <code>HttpService</code> del <code>login.action</code> regresa con el resultado de la validación del usuario, si la validación es verdadera y el tipo de usuario es profesor, se carga el

	menú del profesor, lo mismo para cada tipo de usuario y si es falsa envía un mensaje de error.
--	--

4.4.1.2 Middleware del servidor

La **Tabla 4.2** describe la *action* principal para el módulo de Login.

Tabla 4.2 Action login.action

Action: login.action	
Descripción:	Valida el correo y contraseña del usuario.
Atributos:	Usuario, tipo, autorización, sesión y resultado
Método:	log_in()
Parámetros que recibe:	-
Regresa:	Autorizacion, <i>SUCCESS</i>
Descripción:	Se comunica con la base de datos a través de la clase UsuarioDAO la cual nos devuelve un usuario en caso de que el correo exista en la base de datos y valida la contraseña, regresando el tipo de usuario y una autorización que puede ser verdadera en caso de que el correo y la contraseña sean válidos.

4.4.2 Módulo del Administrador

Los componentes y clases utilizados para este módulo se muestran tanto para la aplicación RIA del cliente como para el *middleware* del servidor.

4.4.2.1 RIA del cliente

La **Tabla 4.3** describe el componente principal para el alta, borrado y modificación de usuarios.

Tabla 4.3 Componente FormularioRegistro.mxml

Componente: FormularioRegistro.mxml	
Descripción:	Presenta el formulario que permite introducir los campos con la información del usuario, para registrarlo en el sistema, así como para actualizarlo y borrarlo.
Atributos:	-
Método:	verResultados()
Parámetros que recibe:	ResultEvent
Regresa:	-
Descripción:	Este método se manda a llamar una vez que el <i>HttpService</i> del registrarUsuario.action regresa con el resultado del registro, el cual es verdadero en caso de que el registro se haya realizado con éxito y en caso contrario es falso.
Método:	desplegarUsuario()
Parámetros que recibe:	ResultEvent
Regresa:	-
Descripción:	Este método se manda llamar una vez que el <i>HttpService</i> del buscarUsuario.action regresa con la información del usuario buscado para poder posteriormente borrar o actualizarse.
Método:	desplegarRes()
Parámetros que recibe:	ResultEvent
Regresa:	-
Descripción:	Este método se manda a llamar una vez que el <i>HttpService</i> del actualizarUsuario.action y el eliminarUsuario.action regresa con el

	resultado de la actualización o eliminación, el cual es verdadero en caso de que la actualización o eliminación se haya realizado con éxito y en caso contrario es falso. .
--	---

4.4.2.2 Middleware del servidor

Las **Tablas 4.4, 4.5, 4.6 y 4.7** describen los *actions* principales para el módulo del Administrador.

Tabla 4.4 Action registrarUsuario.action

Action: registrarUsuario.action	
Descripción:	Registra la información del usuario en la base de datos.
Atributos:	Usuario, tipo y resultado
Método:	registrarUsuario()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Crea un objeto con la información del usuario de acuerdo a su tipo y se comunica con la base de datos a través de la clase UsuarioDAO y su método registraUsuario(Usuario) el cual registra al usuario.

Tabla 4.5 Action buscarUsuario.action

Action: buscarUsuario.action	
Descripción:	Busca al usuario a través de su correo en la base de datos.
Atributos:	Usuario, tipo y resultado
Método:	buscarUsuario()
Parámetros que recibe:	-
Regresa:	Usuario, tipo, resultado, <i>SUCCESS</i>
Descripción:	Busca el correo del usuario en la base de datos, si éste existe envía un resultado verdadero, el tipo y la información del usuario.

Tabla 4.6 Action actualizarUsuario.action

Action: actualizarUsuario.action	
Descripción:	Actualiza al usuario en la base de datos.
Atributos:	Usuario
Método:	actualizarUsuario()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Actualiza al usuario en la base de datos de acuerdo a la información contenida en el objeto usuario y envía un resultado verdadero si se actualizó la información.

Tabla 4.7 Action eliminarUsuario.action

Action: eliminarUsuario.action	
Descripción:	Elimina al usuario en la base de datos.
Atributos:	Usuario
Método:	eliminarUsuario()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Elimina al usuario en la base de datos a través de su correo y envía un resultado verdadero si fue eliminado.

4.4.3 Módulo del Profesor

Los componentes y clases utilizados para este módulo se muestran tanto para la aplicación RIA del cliente como para el *middleware* del servidor.

4.4.3.1 RIA del cliente

La **Tabla 4.8** describe el componente y métodos principales para la creación de cursos.

Tabla 4.8 Componente NuevoCurso.mxml

Componente: NuevoCurso.mxml	
Descripción:	Muestra la interfaz que permite la creación y edición del curso, así como las opciones de: Guardar Tema, Subir Multimedia, Subir Video del profesor, Agregar Pregunta, Crear Examen, Recomendar, Previsualizar, Registrar y Terminar Curso.
Atributos:	contenedorActual, preguntaActual, opcionesXML, respuestasXML, fvCmp, rcCmp, saCmp, llbCmp, omCmp, cadenaResp, misPreguntas, collectionXML, nombreCurso, ruta, xml, valor, idCurso, rutaInfo, tipoInfo, videoXML, nodoVideo, editarPreg, recomendacion.
Método:	cargaContenido()
Parámetros que recibe:	Event
Regresa:	-
Descripción:	Este método se manda a llamar cuando se selecciona un componente prefabricado para la edición de un tema, los tipos de componentes son: solo texto, título doble texto, título texto, solo título y título subtítulo, este método carga al componente seleccionado.
Método:	crearNuevoTema()
Parámetros que recibe:	ResultEvent
Regresa:	-
Descripción:	Este método crea una ventana emergente, que es un componente en el que el usuario introduce el nombre del tema.
Método:	manejador()
Parámetros que recibe:	ItemClickEvent
Regresa:	-
Descripción:	Este método se manda a llamar cuando se selecciona una de las siguientes opciones: Guardar Tema, Multimedia, Previsualizar, Subir Video, Registrar, Pregunta, Examen, Recomendar y Terminar Curso. Opción: Guardar Tema , se envía al servidor el meta datos con la estructura del curso y el meta datos con la información del tema, así como el contenido del tema (en caso de: título texto, título doble texto, y solo texto) a través de guardarCurso.action, guardarInfo.action y guardarTexto.action. Opción: Multimedia , se crea una ventana emergente, en la que el profesor selecciona los multimedia y éstos son guardados en el servidor. Opción: Previsualizar , se carga el componente que permite visualizar el curso. Opción: Subir Video , se crea una ventana emergente en la que el profesor puede seleccionar el video explicativo del curso y el texto del video para ser guardados en el servidor. Opción: Registrar , se registra el curso a través de completarCurso.action para que el alumno pueda inscribirse. Opción: Pregunta , se muestran los tipos de pregunta que pueden ser agregados. Opción: Examen , se muestran las preguntas que se han creado para que el profesor las seleccione y cree el examen. Opción: Recomendar , se muestra el formulario para agregar las recomendaciones de libros, páginas y ejercicios. Por último la Opción: Terminar Curso , se inicializan las variables necesarias para la

	creación de un nuevo curso.
Método:	obtenerResp()
Parámetros que recibe:	CloseEvent
Regresa:	-
Descripción:	Obtiene la respuesta del profesor acerca de si se quiere o no conectar al curso cuando elige la opción de Previsualizar.
Método:	resConHttp()
Parámetros que recibe:	Event
Regresa:	-
Descripción:	Este método se manda a llamar una vez que el HttpService del enLinea.action regresa con el resultado: verdadero si se pudo conectar al curso y falso si no se pudo conectar al curso.
Método:	mandaXMLCurso()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Envía el XML con la estructura del curso al servidor a través del guardarCurso.action para que sea guardado.
Método:	mandaXMLInfo()
Parámetros que recibe:	XML
Regresa:	-
Descripción:	Crea la estructura del XML con la información del componente prefabricado que se seleccionó para el tema y envía el contenido del tema y el XML al servidor.
Método:	cargarTema()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Obtiene el tema seleccionado y carga su contenido para poder editarlo, accediendo al archivo de meta datos que contiene la información del tema.
Método:	eliminar()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Obtiene el tema seleccionado y lo elimina del meta datos que contiene la estructura del curso.

La **Tabla 4.9** describe el componente principal para la creación y edición de una nueva pregunta.

Tabla 4.9 Componente EditarPregunta.mxml

Componente: EditarPregunta.mxml	
Descripción:	Muestra al componente de acuerdo al tipo de pregunta seleccionado, el tipo de pregunta puede ser Falso Verdadero, Opción Múltiple, Llenado en Blanco, Selección de Área y Relación de Columnas, así como las opciones de: Guardar pregunta y Salir.
Atributos:	nombrePregunta, tituloPregunta, rutaPregunta, xml, rutaCurso, pregXML.
Método:	guardar()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Crea el XML con la estructura de la pregunta dependiendo del tipo de pregunta seleccionada y obtiene el número de pregunta a través del obtNumPreguntas.action para después ser guardado en el servidor.
Método:	resultadoHttp()
Parámetros que recibe:	Event
Regresa:	-

Descripción:	Este método se manda a llamar una vez que el <code>HttpService</code> del <code>obtNumPreguntas.action</code> regresa con el número de pregunta para crear el nombre del XML y envía el XML al servidor a través del <code>guardarPregunta.action</code>
Método:	<code>salir()</code>
Parámetros que recibe:	-
Regresa:	-
Descripción:	Permite salir de la creación de preguntas para regresar a la edición del curso.

La **Tabla 4.10** describe el componente principal que permite subir multimedios al servidor.

Tabla 4.10 Componente PopUpSubirMult.mxml

Componente: PopUpSubirMult.mxml	
Descripción:	Presenta una ventana emergente que permite seleccionar multimedios de tipo imagen (jpg, gnp y gif), video (flv) y audio (mp3) y guardarlos en el servidor.
Atributos:	<code>File</code> , <code>jpgFilter</code> , <code>pngFilter</code> , <code>gifFilter</code> , <code>flvFilter</code> , <code>mptFilter</code> , <code>nodo</code> , <code>tmp</code> .
Método:	<code>archivoSeleccionado()</code>
Parámetros que recibe:	<code>Event</code>
Regresa:	-
Descripción:	Crea el meta dato del multimedia para agregárselo al tema seleccionado y sube el archivo al servidor a través del método <code>upload</code> de la clase <code>FileReference</code> .
Método:	<code>buscarArchivo()</code>
Parámetros que recibe:	-
Regresa:	-
Descripción:	Muestra el navegador de archivos, con los filtros especificados de los archivos: <code>jpgFilter</code> , <code>pngFilter</code> , <code>gifFilter</code> , <code>flvFilter</code> y <code>mptFilter</code> .
Método:	<code>enviarCursoXML()</code>
Parámetros que recibe:	-
Regresa:	-
Descripción:	Envía el meta datos del curso actualizado con la información de los multimedios agregados.

La **Tabla 4.11** describe el componente principal que permite agregar recomendaciones en cuanto a libros, páginas de Internet y ejercicios por cada tema seleccionado.

Tabla 4.11 Componente RecomendacionTema.mxml

Componente: RecomendacionTema.mxml	
Descripción:	Muestra los formularios que permiten agregar la información referente a libros, páginas de Internet y ejercicios, generando las etiquetas XML correspondientes al tema seleccionado.
Atributos:	<code>oXML</code> , <code>pXML</code> , <code>eXML</code> , <code>recomXML</code> .
Método:	<code>anadir()</code>
Parámetros que recibe:	-
Regresa:	-
Descripción:	Crea la etiqueta libro con los atributos correspondientes a la información añadida por el profesor.
Nota:	Los métodos <code>anadirPagina()</code> y <code>anadirEj()</code> son similares al anterior.
Método:	<code>guardar()</code>
Parámetros que recibe:	-

Regresa:	-
Descripción:	Envía el meta datos del curso actualizado con la información de las recomendaciones agregadas.

La **Tabla 4.12** describe el componente principal que permite crear una evaluación a partir de las preguntas que ya han sido agregadas.

Tabla 4.12 Componente MisPreguntas.mxml

Componente: MisPreguntas.mxml	
Descripción:	Permite seleccionar preguntas de la lista para crear una nueva evaluación, guardar el archivo de meta datos generado en el servidor o incluso previsualizarla antes de ser guardada.
Atributos:	examXML, ruta, xml, rutaCurso, nombreCurso.
Método:	inicio()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Llama al HttpService de preguntasProf.action para que regrese la lista de preguntas que el profesor ha agregado a ese curso.
Método:	guardar()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Envía al servidor el meta datos de la evaluación que ha sido generado al seleccionar las preguntas.
Método:	previsualizar()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Carga el componente de evaluación para que muestre las preguntas de acuerdo al archivo de meta datos generado, no es necesario que el archivo haya sido guardado en el servidor.

La **Tabla 4.13** describe el componente prefabricado que permite agregar texto al tema.

Tabla 4.13 Componente Texto.mxml

Componente: Texto.mxml	
Descripción:	Este componente es el que se carga al seleccionar el tipo de componente prefabricado: solo texto y permite agregar y editar el contenido del tema.
Nota:	Los componentes prefabricados: título doble texto, título texto, solo título y título subtítulo son similares al anterior.

La **Tabla 4.14** describe el componente principal que permite agregar una pregunta del tipo llenado en blanco.

Tabla 4.14 Componente LlenadoBlanco.mxml

Componente: LlenadoBlanco.mxml	
Descripción:	Muestra el formulario que permite ingresar el título de la pregunta y sus opciones y respuestas, creando un archivo de meta datos por pregunta.
Atributos:	oXML.
Método:	anadir()
Parámetros que recibe:	-
Regresa:	-

Descripción:	Crea la etiqueta opción para agregar las opciones que serán guardadas posteriormente a través del método guardar() del componente EditarPregunta.xml.
Nota:	Los componentes de los tipos de preguntas: Falso Verdadero, Opción Múltiple, Selección de Área y Relación de Columnas son similares al anterior.

La **Tabla 4.15** describe el componente principal para la visualización y eliminación de los cursos creados.

Tabla 4.15 Componente VerCursos.xml

Componente: VerCursos.xml	
Descripción:	Muestra la una lista con el nombre de los cursos que el profesor ha creado obteniéndolos a través del HttpService del cursosProfesor.action, para su posterior edición o eliminación.
Atributos:	idCurso, rutaXML.
Método:	cargaEdicionCurso()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Este método se manda a llamar cuando se da doble click en el nombre del tema, el cual envía una solicitud HttpService para obtener el meta datos con la estructura del curso.
Método:	handleCursoXML()
Parámetros que recibe:	ResultEvent
Regresa:	-
Descripción:	Este método se manda a llamar cuando se carga el meta datos con la estructura del curso, el cual guarda el meta datos y carga la interfaz de edición del curso que es el mismo componente de NuevoCurso.xml solo que con la estructura del curso ya cargada.
Método:	eliminar()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Este método se manda a llamar cuando se presiona la tecla "Esc" para eliminar un curso mandando llamar al HttpService de eliminaCurso.action y enviándole el id del curso.

4.4.3.2 Middleware del servidor

Las **Tablas 4.16, 4.17, 4.18, 4.19, 4.20, 4.21, 4.22, 4.23 y 4.24** describen las *actions* principales para el módulo del Profesor.

Tabla 4.16 Action guardarCurso.action

Action: guardarCurso.action	
Descripción:	Guarda el meta datos con la estructura del curso en el servidor.
Atributos:	Curso, xml, sesión y resultado
Método:	guardaCurso()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Guarda en la base de datos el registro del curso que creó el profesor y escribe el archivo de meta datos en la ruta especificada.

Tabla 4.17 Action guardarInfo.action

Action: guardarInfo.action	
Descripción:	Guarda el meta datos con la estructura del tema en el servidor.
Atributos:	RutaInfo, xml, sesión y resultado
Método:	guardaInfoXML()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Escribe el archivo de meta datos con la información de la estructura del tema en la ruta especificada.

Tabla 4.18 Action guardarTexto.action

Action: guardarTexto.action	
Descripción:	Guarda el archivo de texto en el servidor.
Atributos:	RutaTexto, valor, sesión y resultado
Método:	guardaInfoXML()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Escribe el archivo de texto con el contenido del tema en la ruta especificada.

Tabla 4.19 Action cursosProfesor.action

Action: cursosProfesor.action	
Descripción:	Obtiene los cursos creados por el profesor.
Atributos:	Lista de cursos, resultXML y sesión.
Método:	obtenerCursos()
Parámetros que recibe:	-
Regresa:	Cursos, <i>SUCCESS</i>
Descripción:	Accede a la base de datos para obtener todos los cursos que ha creado el profesor a través de la clase CursoDAO y el método obtenerCursosProf(correo), le envía su correo y arma un archivo de meta datos con el resultado de los cursos.

Tabla 4.20 Action eliminaCurso.action

Action: eliminaCurso.action	
Descripción:	Elimina el curso por su id.
Atributos:	IdCurso, resultado.
Método:	eliminarCurso()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Accede a la base de datos a través de la clase CursoDAO y el método eliminarCursoById(id), le envía el id del curso a eliminar, obteniendo el resultado de la operación.

Tabla 4.21 Action completarCurso.action

Action: completarCurso.action	
Descripción:	Cambia el estado del curso a terminado para que el estudiante pueda inscribirse al curso.
Atributos:	Curso, sesion.
Método:	completarCurso()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Accede a la base de datos a través de la clase CursoDAO y el método completaCurso(ruta), le envía la ruta del curso para que lo actualice en

	la base de datos, obteniendo el resultado de la operación.
--	--

Tabla 4.22 Action enLinea.action

Action: enLinea.action	
Descripción:	El usuario se conecta al curso que esté visualizando para interactuar con los demás participantes.
Atributos:	Curso, sesion.
Método:	conectar()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Actualiza el estado del profesor o estudiante a: conectado.

Tabla 4.23 Action guardarPregunta.action

Action: guardarPregunta.action	
Descripción:	Guarda la pregunta en la base de datos y el archivo de meta datos en el servidor.
Atributos:	Pregunta, sesión, xml.
Método:	guardarPregunta()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	Accede a la base de datos a través de la clase PreguntaDAO y el método guardaPregunta(pregunta), le envía el objeto pregunta y lo guarda en la base de datos, por último escribe el meta datos en el servidor.

Tabla 4.24 Action preguntasProf.action

Action: preguntasProf.action	
Descripción:	Obtiene la lista de preguntas que el profesor ha agregado a ese curso.
Atributos:	Pregunta, sesión.
Método:	obtenerPreguntas()
Parámetros que recibe:	-
Regresa:	preg, <i>SUCCESS</i>
Descripción:	Accede a la base de datos a través de la clase PreguntaDAO y el método obtenerPreguntas(correo, ruta), y regresa una lista de preguntas.

4.4.4 Módulo del Alumno

Los componentes y clases utilizados para este módulo se muestran tanto para la aplicación RIA del cliente como para el *middleware* del servidor.

4.4.4.1 Aplicación RIA del cliente

La **Tabla 4.25** describe el componente principal para visualizar la lista de cursos inscritos.

Tabla 4.25 Componente MisCursos.mxml

Componente: MisCursos.mxml	
Descripción:	Muestra la lista de cursos a los que se ha inscrito el alumno, obtenidos a través del HttpService cursosEstudiante.action para poderlos visualizar.
Atributos:	idCurso, rutaXML.
Método:	inicio()

Parámetros que recibe:	-
Regresa:	-
Descripción:	Al cargarse el componente obtiene la lista de cursos inscritos a través del <code>HttpService cursosEstudiante.action</code> .
Método:	<code>cargaCurso()</code>
Parámetros que recibe:	-
Regresa:	-
Descripción:	Inicializa las variables para que el curso sea cargado en el componente <code>VerCurso.mxml</code> y manda llamar al <code>HttpService enLinea.action</code> para conectar al alumno.
Método:	<code>eliminar()</code>
Parámetros que recibe:	-
Regresa:	-
Descripción:	Eliminar la inscripción al curso a través del <code>HttpService eliminaIns.action</code> al cual se le pasa el id del curso.

La **Tabla 4.26** describe el componente principal para la inscripción del curso.

Tabla 4.26 Componente InsCurso.mxml

Componente: InsCurso.mxml	
Descripción:	Muestra una lista con el nombre de todos los cursos que los profesores han registrado, obtenidos a través del <code>HttpService cursosTodos.action</code> para su posterior inscripción.
Atributos:	<code>idCurso</code> .
Método:	<code>iinscribeCurso()</code>
Parámetros que recibe:	-
Regresa:	-
Descripción:	Este método se manda a llamar cuando se da doble click en el nombre del curso, el cual envía al <code>HttpService del inscribirCurso.action</code> el id del curso para registrar la inscripción.

La **Tabla 4.27** describe el componente principal para la visualización del curso.

Tabla 4.27 Componente VerCurso.mxml

Componente: VerCurso.mxml	
Descripción:	A través de una lista que contiene los cursos a los que el alumno se ha inscrito, el alumno al dar doble click sobre el nombre del curso, el sistema carga este componente para la visualización de dicho curso. Este componente contiene otros componentes para la visualización y comunicación del curso como son: <code>Indice</code> , <code>AreaContenido</code> , <code>AreaExplicacion</code> , <code>VerMultimedios</code> , <code>BarraNavegacion</code> , <code>Participantes</code> , <code>Pizarron</code> , <code>Chat</code> , <code>Escritorio</code> , <code>Descarga</code> y <code>Examen</code> .
Atributos:	<code>Curso</code> , <code>índice</code> , <code>ArrayCollection</code> , <code>aTemasDur</code> .
Método:	<code>llenarCurso()</code>
Parámetros que recibe:	<code>ResultEvent</code>
Regresa:	-
Descripción:	Carga el meta datos con la estructura del curso y llena la clase <code>Curso</code> la cual contiene el nombre del curso y una lista de temas, el componente carga el primer tema.

La **Tabla 4.28** describe el componente principal que muestra los temas del curso.

Tabla 4.28 Componente AreaContenido.xml

Componente: AreaContenido.xml	
Descripción:	Es el componente donde se cargan uno por uno los temas que se encuentran en el arreglo de temas de la clase curso.

La **Tabla 4.29** describe el componente principal que muestra los multimedia de cada tema.

Tabla 4.29 Componente VerMultimedios.xml

Componente: VerMultimedios.xml	
Descripción:	Al seleccionar un tema del curso y dar click en el botón Multimedia se carga este componente con la lista de multimedia agregados para poder visualizarlos o escucharlos.
Atributos:	Multimedios, Sound, SoundChannel, Image, VideoDisplay, ruta.
Método:	cargarMultimedia()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Se manda llamar cuando el usuario selecciona un multimedia de la lista, se obtiene la ruta y el tipo de multimedia a cargar para configurar al componente necesario, es decir, si el tipo es imagen se utiliza el componente Image, si es sonido el componente Sound y SoundChannel y si es video se utiliza el componente VideoDisplay.

La **Tabla 4.30** describe el componente principal que muestra y controla la barra de navegación.

Tabla 4.30 Componente BarraNavegacion.xml

Componente: BarraNavegacion.xml	
Descripción:	Este componente permite separar la parte del contenido del curso de la parte de navegación, es decir, cada que se cambia la navegación del curso, se carga y descarga el tema correspondiente en el componente AreaContenido.xml, se puede navegar en el curso a través de la barra de navegación o a través del componente Indice.xml que muestra un árbol con los temas del curso.
Atributos:	correoUsu, rutaInfo, texto, titulo, tituloDobleTexto, tituloTexto, tituloSub, textoTemp, textoTemp2.
Método:	fPrincipio()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Inicializa el índice a cero y carga el primer tema a través del método cargaTema(index).
Método:	cargaTema()
Parámetros que recibe:	i
Regresa:	-
Descripción:	Obtiene el tema en la posición i del arreglo de temas de la clase curso, verifica el tipo de tema que es, es decir, solo texto, título doble texto, título texto, solo título y título subtítulo y lo carga en el componente correspondiente, es decir, TextoV.xml, TituloV.xml, TituloDobleTextoV, TituloSubtituloV y TituloTextoV.
Método:	fAnterior() y fSiguiente()
Parámetros que recibe:	-
Regresa:	-
Descripción:	fAnterior() disminuye el índice y fSiguiente() incrementa el índice, ambos cargan el tema a través del método cargaTema(index).
Método:	fFin()

Parámetros que recibe:	-
Regresa:	-
Descripción:	Obtiene el índice del último tema y lo carga a través del método cargaTema(index).
Método:	fSalir
Parámetros que recibe:	-
Regresa:	-
Descripción:	Desconecta al usuario a través del HttpService desconectaUsuario.action y manda llamar al método fCerrar() para cargar el Menú correspondiente al tipo de usuario.
Método:	fEval()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Carga al componente Examen.mxml para iniciar la evaluación del curso.

La **Tabla 4.31** describe el componente del pizarrón.

Tabla 4.31 Componente Pizarron.mxml

Componente: Pizarron.mxml	
Descripción:	Al seleccionar la pestaña de Comunicación se cargan algunas herramientas colaborativas como la del pizarrón la cual permite enviar a los participantes del curso: líneas, círculos, elipses, rectángulos, texto, etc.
Atributos:	Dibujando, x1, y1, x2, y2, DPcolor, botonSeleccionado, textoArea, bndTexto, selectorCon.
Método:	inicio()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Este componente hace uso del servidor BlazeDS para enviar mensajes entre los participantes del curso, en este método el consumidor se suscribe con el filtro del curso que está visualizando para que le lleguen todos los mensajes que envíen los participantes productores.
Método:	fMousePresionado()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Obtiene la posición inicial (x1, y1) del mouse.
Método:	fMouseArriba()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Verifica la herramienta seleccionada (línea, círculo, etc.), obtiene la posición final (x2, y2) del mouse y envía como productor el mensaje a los participantes del curso con la información de las posiciones inicial y final, el color y la herramienta seleccionada.
Método:	manejadorMsj()
Parámetros que recibe:	MessageEvent
Regresa:	-
Descripción:	Es el método que se manda llamar cuando el productor envía un mensaje a los consumidores, este método obtiene la información del mensaje y dibuja en el lienzo (<i>Canvas</i>) la herramienta seleccionada en la posición y color estipulado.

La **Tabla 4.32** describe el componente del Chat.

Tabla 4.32 Componente Chat.mxml

Componente: Chat.mxml	
Descripción:	Al seleccionar la pestaña de Comunicación también se carga la herramienta colaborativa del chat, la cual permite enviar mensajes de texto a los participantes del curso.
Atributos:	SelectorCon.
Método:	inicio()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Este componente hace uso del servidor BlazeDS para enviar mensajes entre los participantes del curso, en este método el consumidor se suscribe con el filtro del curso que está visualizando para que le lleguen todos los mensajes que envíen los participantes productores.
Método:	enviar()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Arma el mensaje de texto y lo envía como productor a los consumidores del curso.
Método:	manejadorMsj()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Obtiene la información del mensaje y lo muestra al participante.

La **Tabla 4.33** describe el componente del Escritorio.

Tabla 4.33 Componente Escritorio.mxml

Componente: Escritorio.mxml	
Descripción:	Al seleccionar la pestaña de Comunicación también se carga la herramienta colaborativa del escritorio, la cual permite mostrar el escritorio remoto del participante que lo haya seleccionado.
Atributos:	SelectorCon, servidor.
Método:	inicio()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Este componente hace uso del servidor BlazeDS para enviar mensajes entre los participantes del curso, en este método el consumidor se suscribe con el filtro del curso que está visualizando para que le lleguen todos los mensajes que envíen los participantes productores.
Método:	obtenerIP()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Obtiene la dirección IP del participante que quiere compartir su escritorio remoto para enviársela como mensaje a los consumidores, el participante debe tener activado el servidor UltraVNC.
Método:	manejadorMsj()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Obtiene la dirección IP del mensaje y muestra el escritorio remoto de esa dirección IP.

La **Tabla 4.34** describe el componente principal que muestra la evaluación del curso.

Tabla 4.34 Componente Examen.mxml

Componente: Examen.mxml	
Descripción:	Este componente carga las preguntas guardadas en el archivo de meta datos examen.xml en el componente AreaPregunta y a través de su componente BarraNavegacion va recorriendo la evaluación.
Atributos:	...
Método:	llenarExamen()
Parámetros que recibe:	-
Regresa:	-
Descripción:	Obtiene el archivo de meta datos del servidor y carga la primer pregunta a través del método cargarPregunta(indice).
Método:	cargarPregunta()
Parámetros que recibe:	ind
Regresa:	-
Descripción:	Obtiene la etiqueta pregunta del índice ind y accede a la ruta para poder leer el meta datos de la pregunta correspondiente.
Método:	llenarPregunta()
Parámetros que recibe:	ResultEvent
Regresa:	-
Descripción:	Se manda llamar cuando se ha leído el meta datos de la pregunta, este método se encarga de cargar la información de la pregunta en el componente correspondiente, es decir, FalsoVerdadero, LlenadoBlanco, RelacionColumna, OpcionMultiple y SeleccionArea.

4.4.4.2 Middleware del servidor

Las **Tablas 4.35, 4.36, 4.37, 4.38 y 4.39** describen los *actions* principales para el módulo del Alumno.

Tabla 4.35 Action cursosTodos.action

Action: cursosTodos.action	
Descripción:	Obtiene todos los cursos que han registrado los profesores.
Atributos:	Lista de cursos.
Método:	todos()
Parámetros que recibe:	-
Regresa:	<i>SUCCESS</i>
Descripción:	Accede a la base de datos a través de la clase CursoDAO y obtiene todos los cursos que los profesores han registrado, enviando como resultado una lista con todos los cursos.

Tabla 4.36 Action inscribirCurso.action

Action: inscribirCurso.action	
Descripción:	Inscribe al alumno en el curso que seleccionó por su id.
Atributos:	Id, resultado y sesión.
Método:	inscribirCurso()
Parámetros que recibe:	-
Regresa:	<i>SUCCESS</i>
Descripción:	Obtiene al usuario de la sesión y lo inscribe al curso seleccionado a través de la clase CursoDAO la cual accede a la base de datos.

Tabla 4.37 Action cursosEstudiante.action

Action: cursosEstudiante.action	
Descripción:	Obtiene la lista de cursos a los que el estudiante se ha inscrito.
Atributos:	sesión.
Método:	obtenerCursos()
Parámetros que recibe:	-
Regresa:	<i>SUCCESS</i>
Descripción:	A través de la clase CursoDAO y el método obtenerCursosEst(correo) el cual accede a la base de datos, obtiene la lista de cursos a los que el estudiante se ha inscrito mediante su correo.

Tabla 4.38 Action desconectaUsuario.action

Action: desconectaUsuario.action	
Descripción:	Desconecta al usuario del curso.
Atributos:	correo.
Método:	desconectar()
Parámetros que recibe:	-
Regresa:	Resultado, <i>SUCCESS</i>
Descripción:	A través de la clase UsuarioDAO y los métodos desconectarProf(correo) y desconectarEst(correo) que acceden a la base de datos, se actualiza el curso al que está conectado a nulo.

Tabla 4.39 Action obtenerIP.action

Action: obtenerIP.action	
Descripción:	Obtiene la dirección IP del usuario que comparte su escritorio remoto.
Atributos:	-
Método:	obtenerIP()
Parámetros que recibe:	-
Regresa:	ip, <i>SUCCESS</i>
Descripción:	El método getRemoteAddr del request regresa la dirección ip del usuario.

4.4.5 Módulo del MAS

Para el desarrollo del MAS fue necesario crear un Archivo de Descripción del Agente (*Agent Description File*, ADF por sus siglas en inglés), este archivo de configuración especifica la estructura del agente, la **Tabla 4.40** muestra el ADF del MAS, los planes y metas del agente se especifican en este archivo, para la comunicación con el JADEX-WebBridge es necesario incluir la capacidad *webcap* y el *achivegoal web_request*, cuando se hace una petición *Web* el JADEX-WebBridge dispara la meta *web_request* con lo que se ejecuta el plan *miPlan1* de la clase *SEVirtualPlan*. Se han creado tres metas *achivegoal* que motivan e influyen en el comportamiento del agente, la meta *metrica* que dispara al plan *miPlan2* especificado en la clase *MetricasPlan*, la meta *reconfigurar* que dispara al plan *miPlan3* que ejecuta la clase *ReconfigurarPlan* y la meta *retroalimentar* que dispara al plan *miPlan4* que ejecuta la clase *RetroalimentarPlan*, todas las metas y planes especifican los parámetros que van a utilizar.

Tabla 4.40 Estructura del archivo ADF del sistema: SEVirtualPlan.agent.xml

<pre><agent xmlns="http://jadex.sourceforge.net/jadex" xmlns:xsi="http://www.w3.org/2001/XMLSchema- instance" xsi:schemaLocation="http://jadex.sourceforge.net/jadex- http://jadex.sourceforge.net/jadex- 0.96.xsd"</pre>	<pre><plans> <plan name="miPlan1"> <body class="SEVirtualPlan"/> <trigger> <goal ref="web_request"/> </trigger></pre>
--	---

<pre> name="SEVirtualPlan" package="jadex.sev"> <imports> <import>jadex.*</import> <import>jadex.runtime.*</import> <import>java.util.logging.Level</import> <import>jadex.bridge.onto.*</import> <import>jadex.bridge.application.*</import> </imports> <capabilities> <capability name="webcap" file="WebInteraction"/> </capabilities> <beliefs> <belief name="msg" class="String" exported="true"> <fact>"Welcome to Jadex, desde SEVirtual Plan."</fact> </belief> </beliefs> <goals> <achievegoal name="web_request"> <assignto ref="webcap.web_request"/> <parameter name="request" class="jadex.bridge.onto.Request"> <assignto ref="webcap.web_request.request"/> </parameter> <parameter name="response" class="jadex.bridge.onto.Response" direction="out"> <assignto ref="webcap.web_request.response"/> </parameter> <parameter name="type" class="String"> <assignto ref="webcap.web_request.type"/> </parameter> </achievegoal> <achievegoal name="metrica"> <parameter name="tema" class="String"/> <parameter name="escribir" class="String"/> <parameter name="idCurso" class="String"/> <parameter name="correo" class="String"/> <parameter name="nombrePreg" class="String"/> <parameter name="indicePreg" class="String"/> <parameter name="resultado" class="String"/> </achievegoal> <achievegoal name="reconfigurar"> <parameter name="idCurso" class="String"/> <parameter name="resultado" class="String"/> </achievegoal> <achievegoal name="retroalimentar"> <parameter name="idMetrica" class="String"/> <parameter name="resultado" class="String"/> </achievegoal> </goals> </pre>	<pre> </plan> <plan name="miPlan2"> <parameter name="tema" class="String"> <goalmapping ref="metrica.tema"/> </parameter> <parameter name="escribir" class="String"> <goalmapping ref="metrica.escribir"/> </parameter> <parameter name="idCurso" class="String"> <goalmapping ref="metrica.idCurso"/> </parameter> <parameter name="correo" class="String"> <goalmapping ref="metrica.correo"/> </parameter> <parameter name="nombrePreg" class="String"> <goalmapping ref="metrica.nombrePreg"/> </parameter> <parameter name="indicePreg" class="String"> <goalmapping ref="metrica.indicePreg"/> </parameter> <parameter name="resultado" class="String"> <goalmapping ref="metrica.resultado"/> </parameter> <body class="MetricasPlan"> <trigger> <goal ref="metrica"/> </trigger> </body> </plan> <plan name="miPlan3"> <parameter name="idCurso" class="String"> <goalmapping ref="reconfigurar.idCurso"/> </parameter> <parameter name="resultado" class="String"> <goalmapping ref="reconfigurar.resultado"/> </parameter> <body class="ReconfigurarPlan"> <trigger> <goal ref="reconfigurar"/> </trigger> </body> </plan> <plan name="miPlan4"> <parameter name="idMetrica" class="String"> <goalmapping ref="retroalimentar.idMetrica"/> </parameter> <parameter name="resultado" class="String"> <goalmapping ref="retroalimentar.resultado"/> </parameter> <body class="RetroalimentarPlan"> <trigger> <goal ref="retroalimentar"/> </trigger> </body> </plan> </plans> <configurations> <configuration name="default"> </configuration> </configurations> </agent> </pre>
--	--

La clase SEVirtualPlan es la que se ejecuta cuando se envía una petición *Web*, la **Tabla 4.41** muestra un fragmento del código de esta clase, SEVirtualPlan se encarga de disparar la meta correspondiente a la petición que debe alcanzar el agente, si el tipo de petición es *métricas* el

agente guardará las métricas enviadas durante la evaluación en la base de datos, si el tipo de petición es *reconfigurar* el agente reconfigurará el curso en base a las métricas guardadas en la base de datos durante la evaluación y si el tipo de petición es *retroalimentar* el agente conformará la retroalimentación dinámica que se dará al estudiante así como las recomendaciones.

Tabla 4.41 Planes de SEVirtualPlan.java

```

if (tipo.equals("metricas")) {
    meta = createGoal("metrica");
    meta.getParameter("tema").setValue(request.getParameterObject("tema").getValue().toString());
    meta.getParameter("escribir").setValue(request.getParameterObject("escribir").getValue().toString());
    meta.getParameter("idCurso").setValue(request.getParameterObject("idCurso").getValue().toString());
    meta.getParameter("correo").setValue(request.getParameterObject("correo").getValue().toString());
    meta.getParameter("nombrePreg").setValue(request.getParameterObject("nombrePreg").getValue().toString());
    meta.getParameter("indicePreg").setValue(request.getParameterObject("indicePreg").getValue().toString());
    meta.getParameter("resultado").setValue(resultado);
    dispatchSubgoalAndWait(meta);
    resultado = meta.getParameter("resultado").getValue().toString();
} else if (tipo.equals("reconfigurar")) {
    meta = createGoal("reconfigurar");
    meta.getParameter("idCurso").setValue(request.getParameterObject("idCurso").getValue().toString());
    meta.getParameter("resultado").setValue(resultado);
    dispatchSubgoalAndWait(meta);
    resultado = meta.getParameter("resultado").getValue().toString();
} else if (tipo.equals("retroalimentar")) {
    meta = createGoal("retroalimentar");
    meta.getParameter("idMetrica").setValue(request.getParameterObject("idMetrica").getValue().toString());
    meta.getParameter("resultado").setValue(resultado);
    dispatchSubgoalAndWait(meta);
    resultado = meta.getParameter("resultado").getValue().toString();
}

```

La **Tabla 4.42** muestra un fragmento del código de la clase *MetricasPlan*, en el cual se observa que las métricas son registradas en la base de datos.

Tabla 4.42 Plan MetricasPlan.java para registro en la base de datos.

```

MetricaDAO metDAO = new MetricaDAO();
int idMet = metDAO.registrarMetrica(id, correo, ruta);
System.out.println("ID MET: " + idMet);

MetricaPregunta metPreg = new MetricaPregunta();
metPreg.setTemaXml(tema);
metPreg.setNombrePregunta(nombre);
metPreg.setIndicePregunta(Integer.parseInt(indice));
metDAO.registrarMetricaPregunta(idMet, metPreg);
getParameter("resultado").setValue(String.valueOf(idMet));

```

La **Tabla 4.43** muestra un fragmento del código de la clase *ReconfigurarPlan*, en el cual se observa que se obtiene la lista de métricas de la base de datos y a partir de esa lista se arma el meta datos del curso reconfigurado.

Tabla 4.43 Plan de reconfiguración: ReconfigurarPlan.java

```

MetricaDAO metDAO = new MetricaDAO();
List lista = metDAO.obtenerTemasDif(idMetInt);
...

```

```

String xml= "<sevirtual><curso nombre=\""+nombreCurso+"\">";
DocumentBuilderFactory dbf;
DocumentBuilder db;
InputSource is;
Document doc; NodeList nList; Node nNode; Element eElement;
Iterator it = lista.iterator();
try {
while(it.hasNext()){
String tmpMP = (String)it.next();
dbf = DocumentBuilderFactory.newInstance();
db = dbf.newDocumentBuilder();
is = new InputSource();
is.setCharacterStream(new StringReader(tmpMP));
doc = db.parse(is);
nList = doc.getElementsByTagName("tema");
nNode = nList.item(0);
eElement = (Element) nNode;
rutaTema = eElement.getAttribute("ruta");
if(!rutaTemas.contains(rutaTema)){
rutaTemas.add(rutaTema);
lstTemas.add(tmpMP);
}
}
} catch (Exception e) {
e.printStackTrace();
}

it = lstTemas.iterator();
while(it.hasNext()){
String tmpMP = (String)it.next();
xml+= tmpMP;
}
xml+="</curso></sevirtual>";

```

La **Tabla 4.44** muestra un fragmento del código de la clase *RetroalimentarPlan*, en el cual se observa que se obtiene la lista de métricas de la base de datos y a partir de esa lista se arma la retroalimentación dinámica con el nombre de la pregunta que se equivocó y el tema que se le dificultó al alumno, así como las recomendaciones con los libros, páginas de Internet y ejercicios relacionados con el tema.

Tabla 4.44 Plan de retroalimentación RetroalimentarPlan.java

```

MetricaDAO metDAO = new MetricaDAO();
List lista = metDAO.obtenerTemas(idMetInt);

...
Iterator it = lista.iterator();
while(it.hasNext()){
libros = new ArrayList();
paginas = new ArrayList();
ejercicios = new ArrayList();

MetricaPregunta tmpMP = (MetricaPregunta)it.next();
xml+= "<pregunta>";
xml+= "<nombre>"+tmpMP.getNombrePregunta()+"</nombre>";

try {
DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
DocumentBuilder db = dbf.newDocumentBuilder();
InputSource is = new InputSource();
is.setCharacterStream(new StringReader(tmpMP.getTemaXml()));
Document doc = db.parse(is);

NodeList nList = doc.getElementsByTagName("tema");
Node nNode = nList.item(0);
Element eElement = (Element) nNode;
xml+= "<tema>"+eElement.getAttribute("nombre")+"</tema>";

...

```

```

xml+= "<libro>";
Iterator it2 = libros.iterator();
while(it2.hasNext()){
    String tmpStr = (String)it2.next();
    xml+= tmpStr;
}
xml+= "</libro>";

xml+= "<pagina>";
Iterator it3 = paginas.iterator();
while(it3.hasNext()){
    String tmpStr = (String)it3.next();
    xml+= tmpStr;
}
xml+= "</pagina>";

xml+= "<ejercicio>";
Iterator it4 = ejercicios.iterator();
while(it4.hasNext()){
    String tmpStr = (String)it4.next();
    xml+= tmpStr;
}
xml+= "</ejercicio>";

} catch (Exception e) {
    e.printStackTrace();
}
xml+= "<indice>"+tmpMP.getIndicePregunta()+"</indice>";

xml+= "</pregunta>";
    
```

Para levantar la plataforma JADEX es necesario ejecutar el siguiente comando: `java jadex.adapter.standalone.Platform`, una vez iniciada se dan de alta los siguientes jars: *webbridge_agent.jar*, *sevirtualplan.jar* y los jars necesarios para ejecutar el Hibernate, por último se instancia el agente Coordinador (*Coordinator.agent.xml*) del JADEX-WebBridge, la **Figura 4.4** muestra la plataforma JADEX con el agente Coordinador ejecutándose.

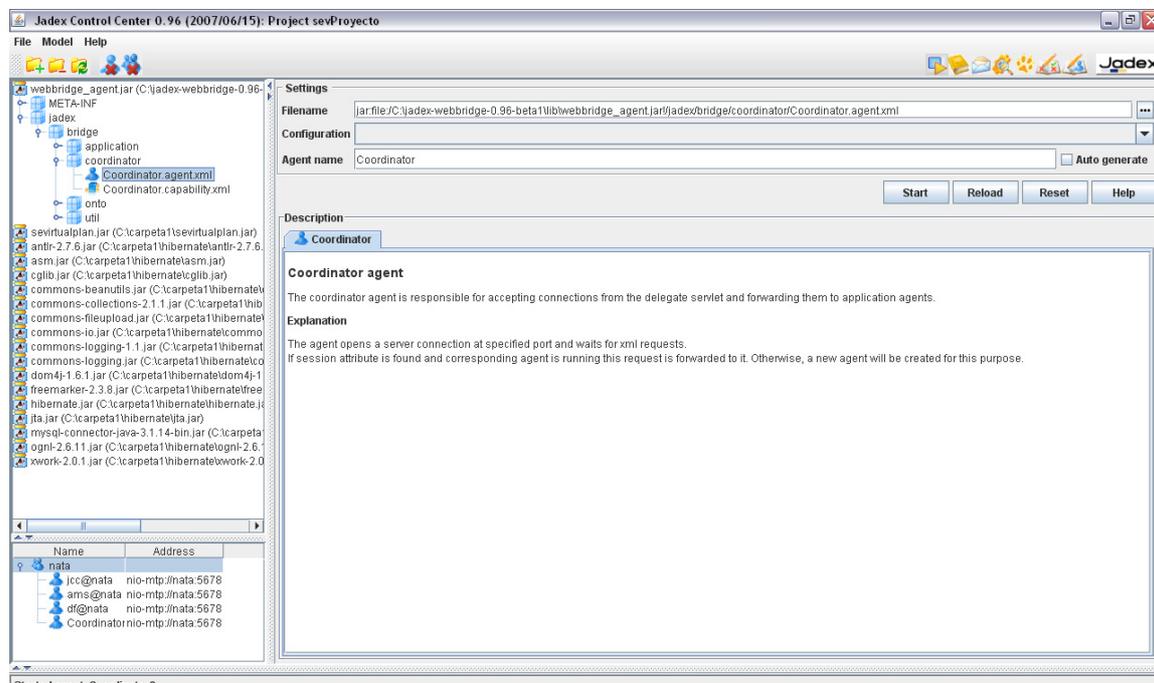


Figura 4.4 Plataforma JADEX

El archivo *sevirtualplan.jar* está formado por el ADF, los planes, el archivo de configuración de Hibernate y las clases ADO, de igual manera es necesario crear una aplicación *Web* dinámica con esos archivos, agregando el jar *webbridge_servlet.jar*, los jars de Hibernate, el archivo de configuración *web.xml* y montarla en el servidor Tomcat 6.0, todo esto es necesario para que el MAS y la comunicación con el JADEX-WebBridge funcione adecuadamente.

4.5 TECNOLOGÍAS DE SOFTWARE

Algunas tecnologías utilizadas para el desarrollo de SEVirtual son las siguientes:

- Programación Orientada a Objetos: Java, JSP's, Servlets, Apache Struts2.
- Programación Orientada a Componentes: Adobe Flex3.
- Programación Orientada a Agentes: JADEX.
- Multimedia.
- Servidores Web: Apache Tomcat 6.0, BlazeDS.
- XML.
- Redes.
- Bases de Datos: MySQL.
- Hibernate, Xalan, Xerces, Action Script.
- UltraVNC

A continuación se describen algunas de ellas.

4.5.1 Apache Struts 2

Struts 2 es un *framework* de código abierto, extensible, útil para crear aplicaciones *Web* empresariales. El *framework* está diseñado para simplificar el ciclo de desarrollo completo, desde la implementación hasta el mantenimiento. Struts 2 está basado en Java y utiliza la distribución 5 o posterior.

Struts 2 cuenta con las siguientes características:

- Permite crear aplicaciones *Web* en Java
- Respuesta dinámica
- Interacción con Bases de Datos, Servicios *Web*, etc.

El *middleware* del servidor proporciona una serie de actions que permiten dar respuesta a los servicios del lado del cliente como son: inscribir curso, guardar curso, eliminar curso, login, registrar usuario, etc.

4.5.2 Adobe Flex 3

Adobe Flex 3 es una *framework* para la creación de aplicaciones RIAs. Sus características principales son:

- Utiliza lenguaje ActionScript 3.0 (orientado a objetos).
- Utiliza MXML, un lenguaje basado en XML.
- SDK de Flex gratis.
- Multiplataforma

- Herramientas de compilación.
- Librerías reutilizables.
- Depurador en línea de comando.
- Generador de documentación.

4.5.3 BlazeDS

BlazeDS es un servidor remoto basado en Java y tecnología *Web* de mensajería que permite a los desarrolladores conectar datos distribuidos y enviarlos en tiempo real a aplicaciones RIA. La integración y evolución de las aplicaciones RIA ha creado la necesidad de mejorar las opciones de conectividad de datos. BlazeDS proporciona a la comunidad de desarrolladores de Adobe en rápido crecimiento, libre acceso a la comunicación remota de gran alcance y tecnologías de mensajería desarrolladas por Adobe.

BlazeDS se utilizó como un servidor para lograr una comunicación entre los participantes de los cursos a través del chat, el pizarrón y el escritorio remoto.

4.5.4 UltraVNC

UltraVNC es un software libre de gran alcance que permite compartir el escritorio remoto de una computadora (a través de la Internet o de una red) a otra. Esta comunicación remota está disponible a través del protocolo HTTP lo cual facilita la comunicación con aplicaciones *Web*.

UltraVNC se utilizó para compartir el escritorio remoto de un participante del curso a los demás participantes, a través del protocolo http y del puerto 5800.

4.6 ESTRUCTURA DEL CURSO

El empleo de meta datos dentro del sistema permite reducir la complejidad de la base de datos, además nos permite organizar la estructura del curso, en una estructura abierta para el profesor, si se quisieran agregar nuevos tipos de temas (solo texto, título doble texto, etc.) a la estructura de los cursos, solo sería necesario crear el meta datos, sin necesidad de modificar la base de datos.

Para ejemplificar el archivo de meta datos que se utiliza para organizar la estructura de un curso de entrenamiento, la **Tabla 4.45** muestra el archivo de meta datos para el curso "Programación Orientada a Objetos", en el cual se observa que los temas se encuentran agrupados por el elemento curso, el cual tiene como atributo el nombre del curso, el elemento tema tiene atributos como el nombre, tipo y ruta del tema, éste a su vez puede agrupar a más temas y así sucesivamente, ya que la estructura que se maneja es abierta.

Cada elemento tema puede agrupar multimedios, con atributos como: tipo, nombre y ruta, esta información es necesaria para el despliegue de los multimedios en la visualización del curso. También puede agrupar recomendaciones de libros, páginas *Web* y ejercicios a través del elemento recomendación, estas recomendaciones son importantes para el MAS el cual se encarga de recomendar al alumno en base a los temas que se le dificultaron una vez evaluado.

Tabla 4.45 Estructura de meta datos para la organización del curso

<sevirtual>

```

<curso nombre="Programacion OO">
  <tema tipo="tituloDobleTexto" nombre="Java" ruta="Programacion
OO//Java//Java.xml">
    <tema tipo="tituloSubtitulo" nombre="Ejemplos" ruta="Programacion
OO//Java//Ejemplos//Ejemplos.xml"/>
    <multimedia tipo=".jpg" nombre="compu.jpg" ruta="Programacion
OO//Java//compu.jpg"/>
    <multimedia tipo=".jpg" nombre="rep.jpg" ruta="Programacion
OO//Java//rep.jpg"/>
    <recomendacion>
      <libro autor="Knuth" nombre="Java Basico" editorial="Mc Graw Hill"/>
      <libro autor="Odd" nombre="Java Avanzado" editorial="Venus"/>
      <pagina nombre="www.javaworld.com"/>
      <pagina nombre="www.tutorialjava.html"/>
      <ejercicio nombre="ejercicios1.pdf"/>
      <ejercicio nombre="programas.gif"/>
    </recomendacion>
  </tema>
  <tema tipo="tituloTexto" nombre="C++" ruta="Programacion OO//C++//C++.xml">
    <multimedia tipo=".mp3" nombre="ser el viento.mp3" ruta="Programacion
OO//C++//ser el viento.mp3"/>
    <recomendacion>
      <libro nombre="C plus plus" autor="Jorge Rosas" editorial="Anaya"/>
      <libro nombre="CPP basico" autor="Jules" editorial="Safiro"/>
      <pagina nombre="www.cpptraining.com"/>
    </recomendacion>
  </tema>
  <tema tipo="soloTexto" nombre="Ada" ruta="Programacion OO//Ada//Ada.xml">
    <multimedia tipo=".jpg" nombre="parar.jpg" ruta="Programacion
OO//Ada//parar.jpg"/>
  </tema>
</curso>
</sevirtual>

```

El elemento tema contiene información de la ruta donde se encuentra el archivo de meta datos con la información del tipo de componente que el profesor seleccionó, la **Tabla 4.46** muestra un ejemplo para el tipo de componente: título doble texto, donde se muestra la información del curso dentro del elemento título y las rutas donde se encuentran los archivos de texto con el contenido del tema.

Tabla 4.46 Estructura de meta datos para el tipo de componente: título doble texto

```

<info>
  <titulo>&lt;TEXTFORMAT LEADING="2"&gt;&lt;P ALIGN="CENTER"&gt;&lt;FONT
FACE="Verdana" SIZE="36" COLOR="#0B333C" LETTERSPACING="0"
KERNING="0"&gt;&lt;B&gt;Java&lt;/B&gt;&lt;/FONT&gt;&lt;/P&gt;&lt;/TEXTFORMAT&gt;
  </titulo>
  <texto ruta="Programacion OO//Java//texto1.txt"/>
  <texto ruta="Programacion OO//Java//texto2.txt"/>
</info>

```

4.7 ESTRUCTURA DEL EXAMEN Y PREGUNTAS

La estructura de la evaluación creada por el profesor también se guarda en un archivo de meta datos, la **Tabla 4.47** muestra un ejemplo de una evaluación, con los nombres de las preguntas y rutas como atributos, en la ruta se encuentra la información de la pregunta dependiendo del tipo de

pregunta seleccionado. Si el profesor quisiera crear una nueva evaluación solo es necesario crear la estructura del meta datos con las etiquetas de las preguntas seleccionadas.

Tabla 4.47 Estructura de meta datos para la evaluación

```
<examen>
  <pregunta nombre="Pregunta 2 "
    ruta="//pere@hotmail.com//Programacion 00//preguntas//pregunta2.xml "/>
  <pregunta nombre="Pregunta 22 "
    ruta="//pere@hotmail.com//Programacion 00//preguntas//pregunta22.xml "/>
  <pregunta nombre="Pregunta 34 "
    ruta="//pere@hotmail.com//Programacion 00//preguntas//pregunta34.xml "/>
  <pregunta nombre="Pregunta 24 "
    ruta="//pere@hotmail.com//Programacion 00//preguntas//pregunta24.xml "/>
  <pregunta nombre="Pregunta 23 "
    ruta="//pere@hotmail.com//Programacion 00//preguntas//pregunta23.xml "/>
</examen>
```

Para la organización de las preguntas también se utilizan los meta datos, una de las ventajas es que si se quisiera agregar un nuevo tipo de pregunta (falso verdadero, opción múltiple, etc.) solo se tendría que crear la estructura del meta datos, sin necesidad de modificar la base de datos. La **Tabla 4.48** muestra un ejemplo de la estructura de meta datos de una pregunta del tipo llenado en blanco, se guarda el título de la pregunta, el tema al que corresponde y las posibles respuestas.

Tabla 4.48 Estructura de meta datos para el tipo de pregunta: llenado en blanco

```
<pregunta>
  <llenadoblanco>
    <titulo>Característica de POO</titulo>
    <tema tipo="tituloTexto" nombre="C++" ruta="Programacion 00//C++//C++.xml">
      <multimedia tipo=".mp3" nombre="ser el viento.mp3" ruta="Programacion
        00//C++//ser el
        viento.mp3"/>
      <recomendacion>
        <libro nombre="C plus plus" autor="Jorge Rosas" editorial="Anaya"/>
        <libro nombre="CPP basico" autor="Jules" editorial="Safiro"/>
        <pagina nombre="www.cpptraining.com"/>
      </recomendacion>
    </tema>
    <respuestas>
      <opcion texto="Herencia"/>
      <opcion texto="Polimorfismo"/>
      <opcion texto="Encapsulación"/>
    </respuestas>
  </llenadoblanco>
</pregunta>
```

4.8 DIAGRAMA ENTIDAD-RELACIÓN

La **Figura 4.5** muestra el diagrama relacional de SEVirtual, el cual está implementado a través del manejador de base de datos MySQL [61]. Se utiliza la generalización para manejar los tres tipos de usuario que existen en el sistema: el profesor, el administrador y el estudiante, el profesor puede crear muchos cursos, un estudiante se puede inscribir a muchos cursos y un curso puede ser inscrito por muchos estudiantes, estas son las relaciones que se muestran en el diagrama relacional y que nos permiten manejar la administración de los usuarios y los cursos. Para la

administración de las evaluaciones se registran las rutas de las preguntas que guarda cada profesor por cada curso y para la administración de las métricas que se envían durante la evaluación del alumno, se utiliza la tabla *Metrica* que lleva un registro de la métrica por evaluación y cada métrica se guarda en la tabla *MetricaPregunta*, la cual tiene información de la métrica a la que pertenece, el nombre e índice de cada pregunta y el meta datos del tema que se le dificultó.

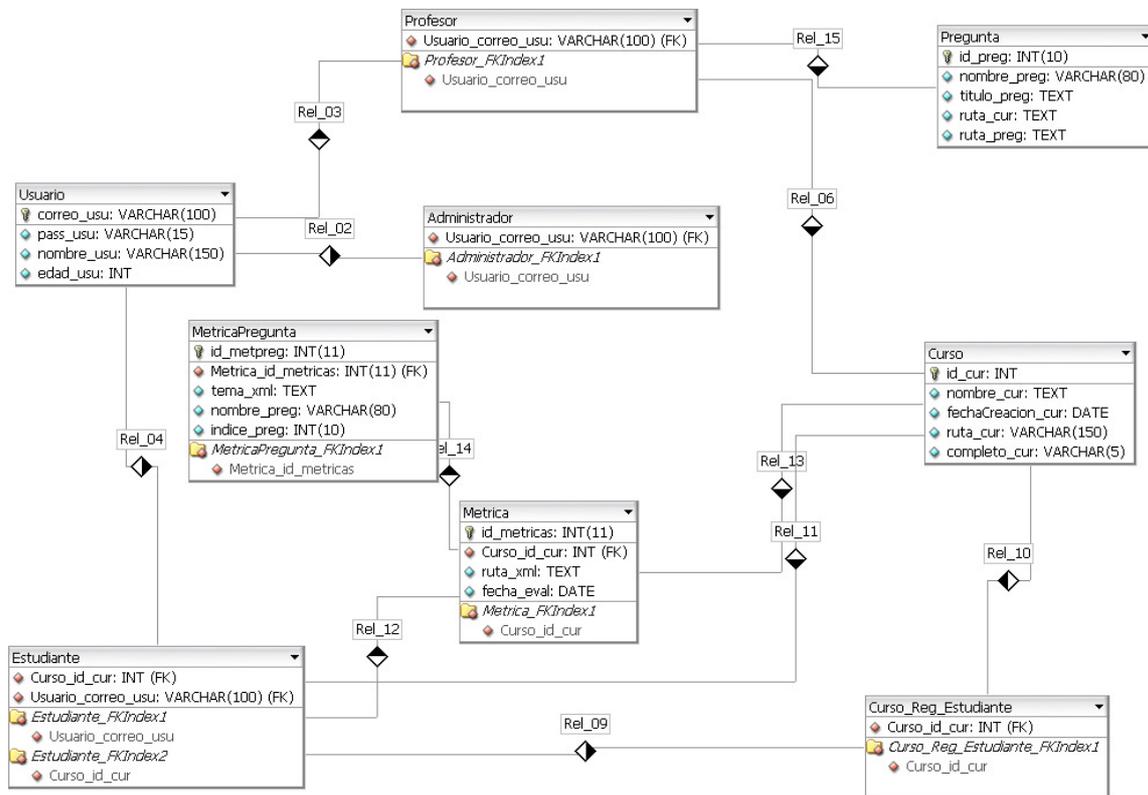


Figura 4.5 Diagrama relacional de SEVirtual

4.9 RESUMEN

En el presente capítulo se abordó la implementación de SEVirtual, presentando los patrones de diseño y las tecnologías utilizadas, además se describieron los componentes principales para la aplicación RIA del cliente y las actions implementados en el middleware del servidor.

También se mostraron los archivos de meta datos que manejan la organización del curso y la evaluación, así como el diagrama entidad relación implementado en MySQL para lograr la administración de los usuarios, cursos, preguntas y métricas que son registradas en la base de datos.

En el siguiente capítulo se presentan algunas pruebas realizadas a SEVirtual y una comparativa de la aplicación con otras aplicaciones comerciales.

CAPÍTULO 5

PRUEBAS Y RESULTADOS DE SEVIRTUAL

En el presente capítulo se abordan los resultados obtenidos de las pruebas realizadas a SEVirtual. Las pruebas que se han realizado muestran la funcionalidad de SEVirtual definiendo un escenario que consta de cinco actores: un administrador, un profesor y tres alumnos. También se muestra la Tabla 5.1 que compara la funcionalidad de SEVirtual con respecto a otras aplicaciones similares.

5.1 PRUEBAS DE SEVIRTUAL

La prueba se realizó en un laboratorio con seis computadoras conectadas de forma concurrente a través de la *Web*: una computadora servidor y cinco computadoras cliente, los actores que participaron fueron: un administrador, un profesor y 3 alumnos.

El escenario fue el siguiente:

- El administrador da de alta un usuario, modifica un usuario y elimina un usuario.
- El profesor da de alta un nuevo curso de entrenamiento.
- El primer alumno navega a través del curso y envía mensajes a través del chat y comparte su escritorio a todos los participantes del curso.
- El segundo alumno navega a través del curso y envía mensajes a través del chat y del pizarrón electrónico.
- El tercer alumno navega a través del curso y contesta la evaluación.

A continuación se detalla el escenario planteado:

Todas las operaciones se realizan de manera concurrente y a través de la *Web*, el administrador realiza sus funciones mientras que el profesor crea un curso y los alumnos interactúan y visualizan el curso Photoshop.

El administrador da de alta un usuario con la siguiente información:

- a. Correo: pedro@hotmail.com
- b. Nombre: Pedro Valderrama
- c. Edad: 23
- d. Tipo de usuario: estudiante
- e. Password: pedrov

La **Figura 5.1** muestra el formulario con los datos del nuevo usuario y la **Figura 5.2** presenta el mensaje de la operación realizada.

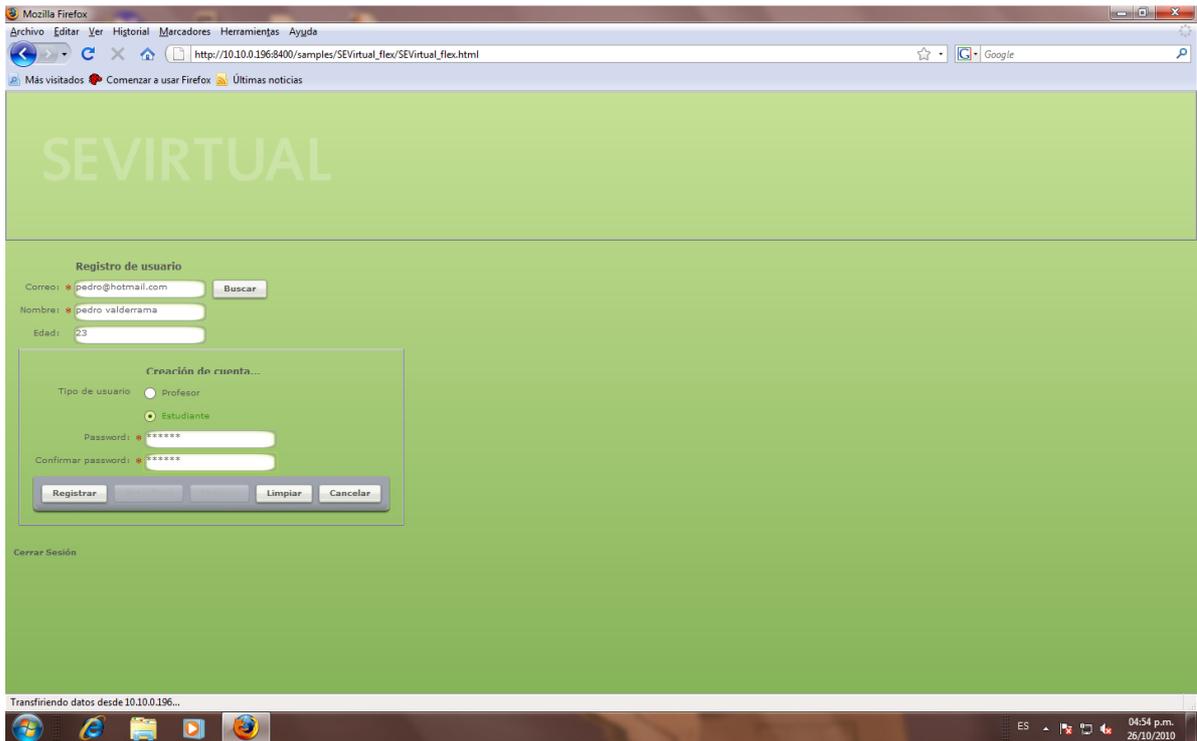


Figura 5.1 Información del nuevo usuario

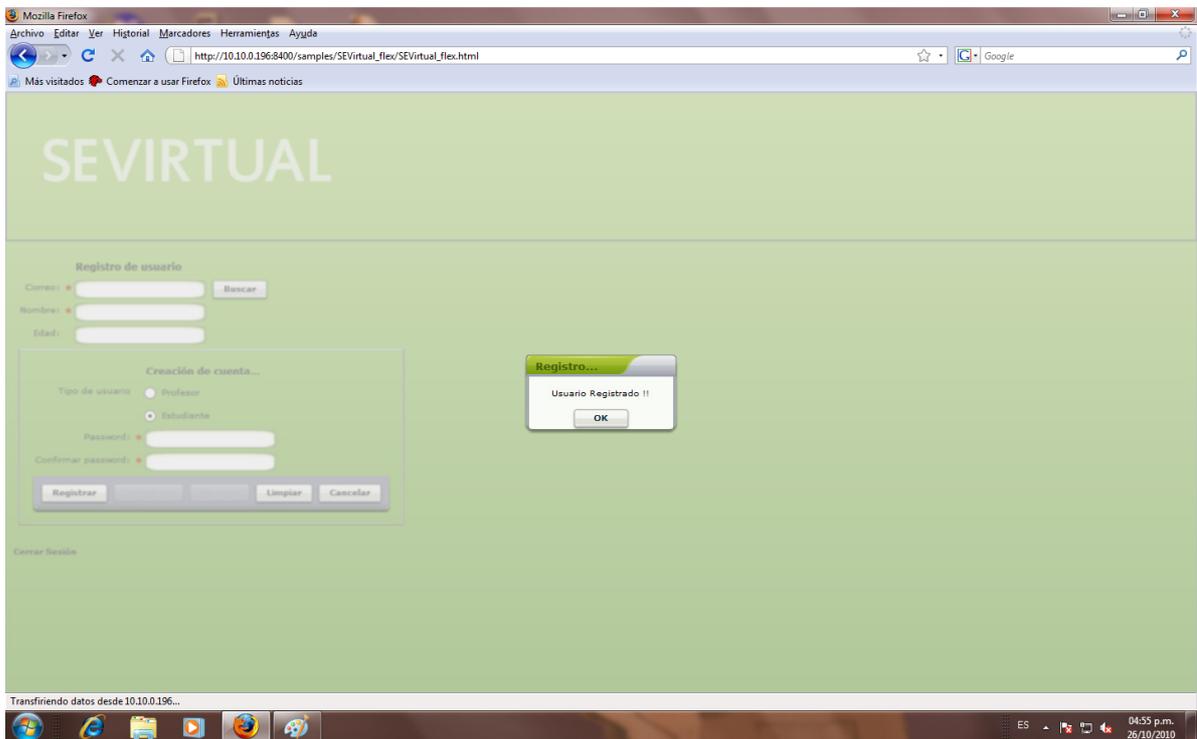


Figura 5.2 Usuario registrado

El profesor se encuentra dando de alta un nuevo curso con el nombre de "Illustrator", la **Figura 5.3** muestra al profesor iniciando sesión dentro del sistema y la **Figura 5.4** creando el nuevo curso.

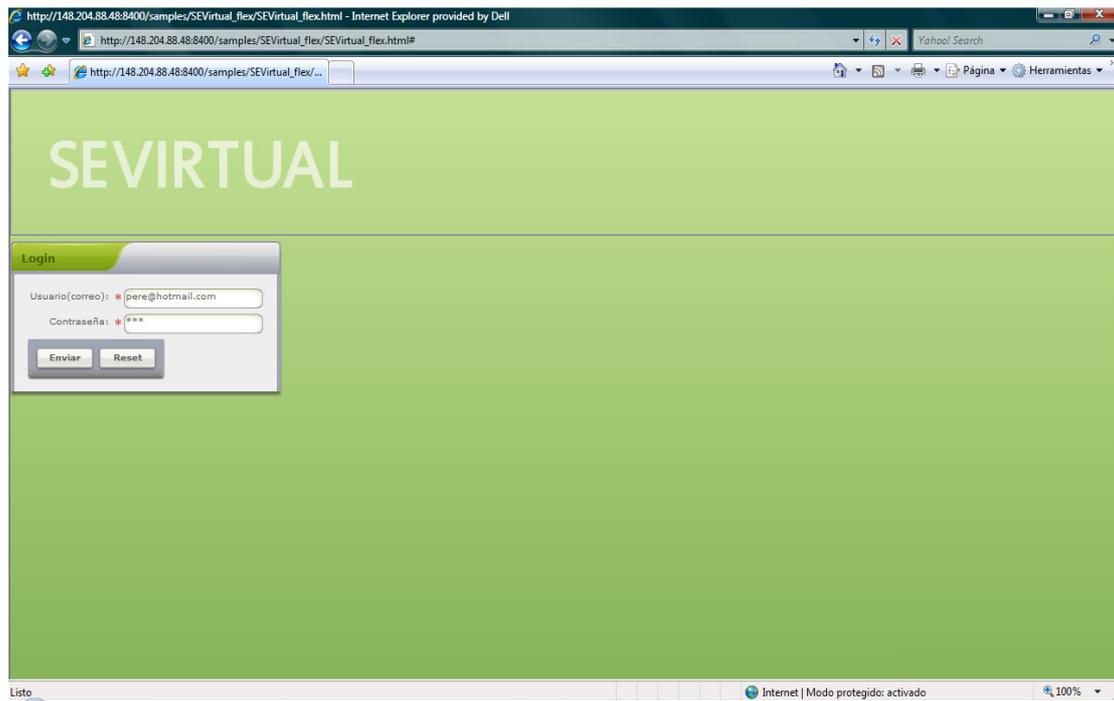


Figura 5.3 Login – profesor

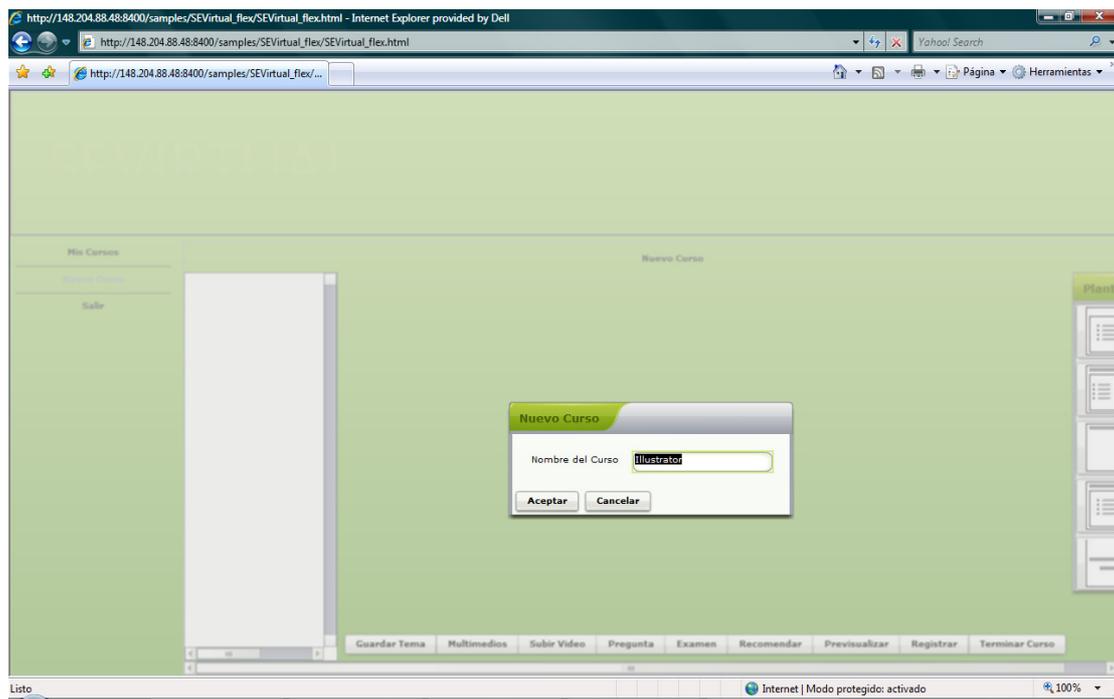


Figura 5.4 Nuevo curso – profesor

El profesor crea tres nuevos temas: “Introducción”, “Edición” y “Líneas” y edita cada uno de ellos, la **Figura 5.5** presenta al profesor agregando el tema “Introducción” y editándolo al seleccionar el componente: “Título doble texto”, por último la **Figura 5.6** muestra el resultado de guardar este tema.

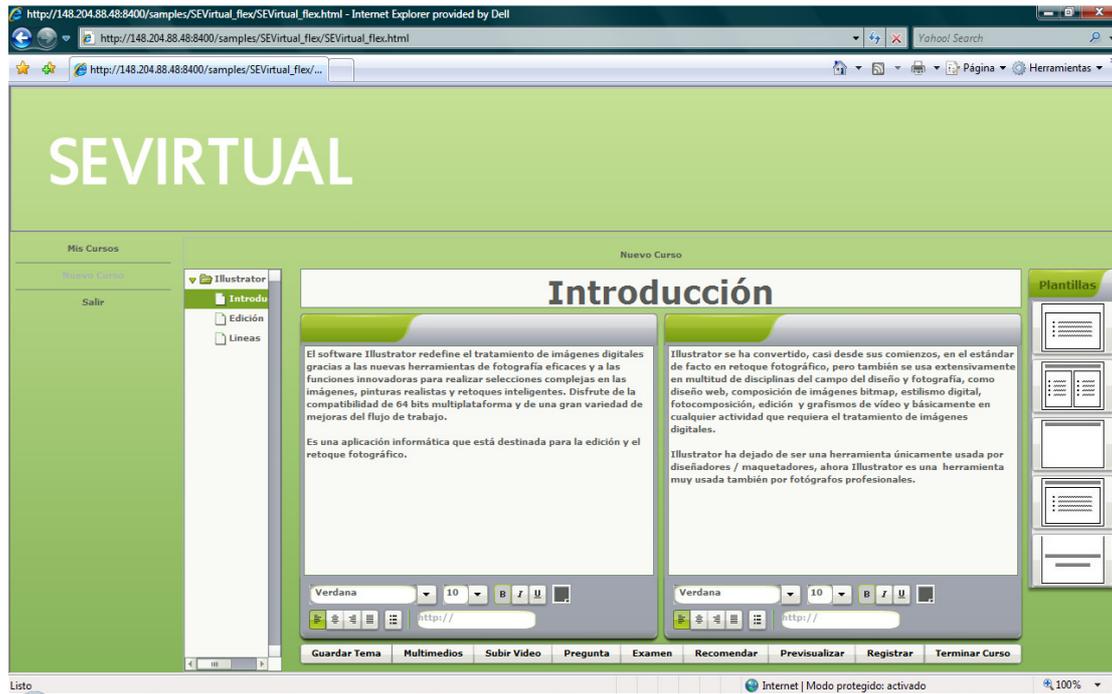


Figura 5.5 Tema: "Introducción" – profesor

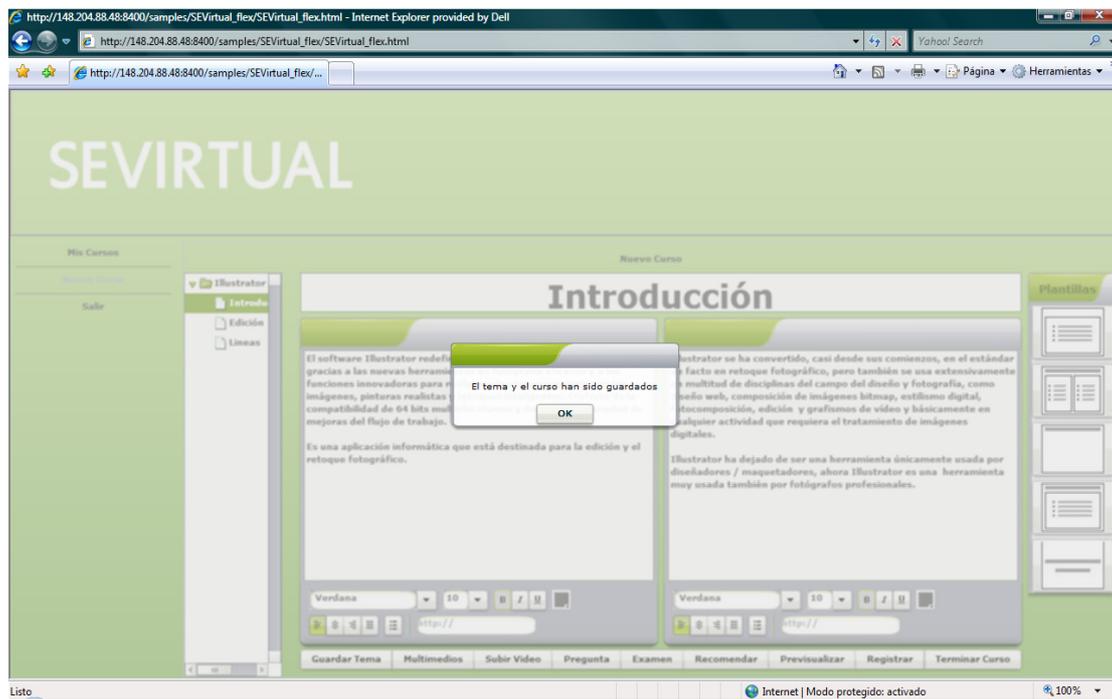


Figura 5.6 Tema "Introducción" guardado – profesor

Por otro lado el primer alumno inicia sesión en el sistema con el correo: pedro@hotmail.com y la contraseña: pedrov, que ya ha sido registrado por el administrador, la **Figura 5.7** despliega la interfaz de inicio de sesión al sistema.

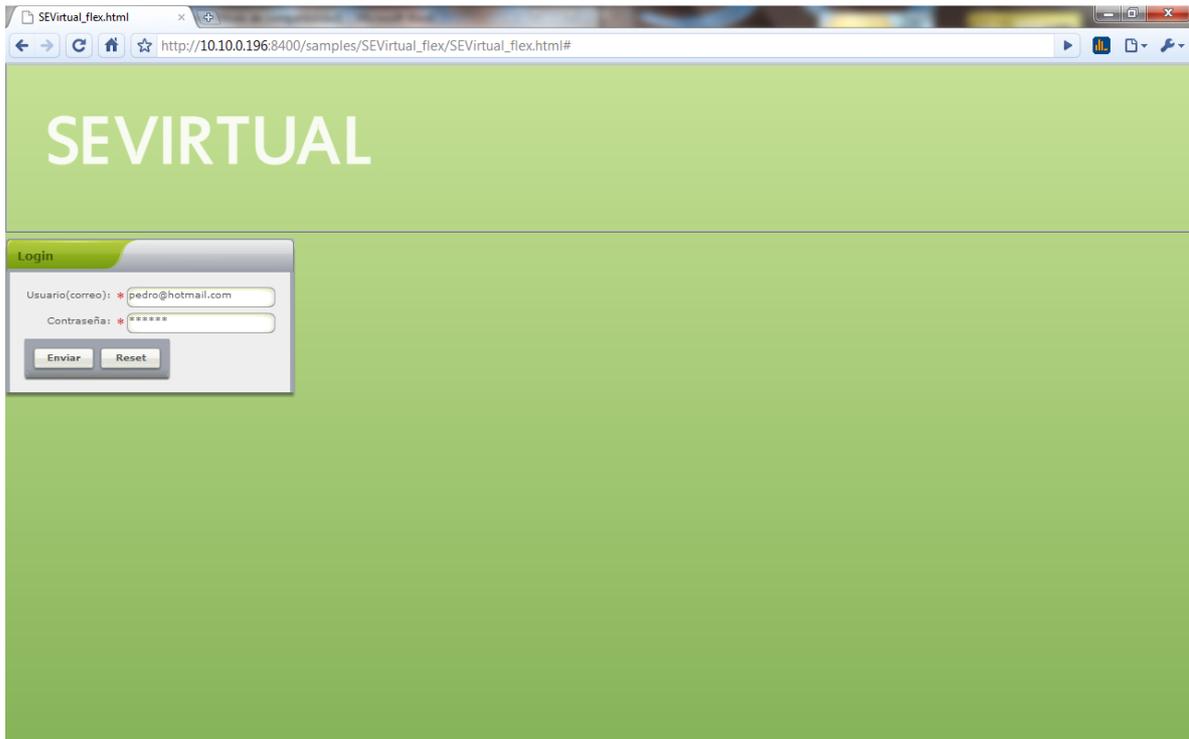


Figura 5.7 Login - primer alumno

El primer alumno se inscribe al curso “Photoshop” y lo visualiza, la **Figura 5.8** presenta cómo el alumno navega a través del curso dando clic en cada tema del índice, en este caso se encuentra en el tema “Introducción”.

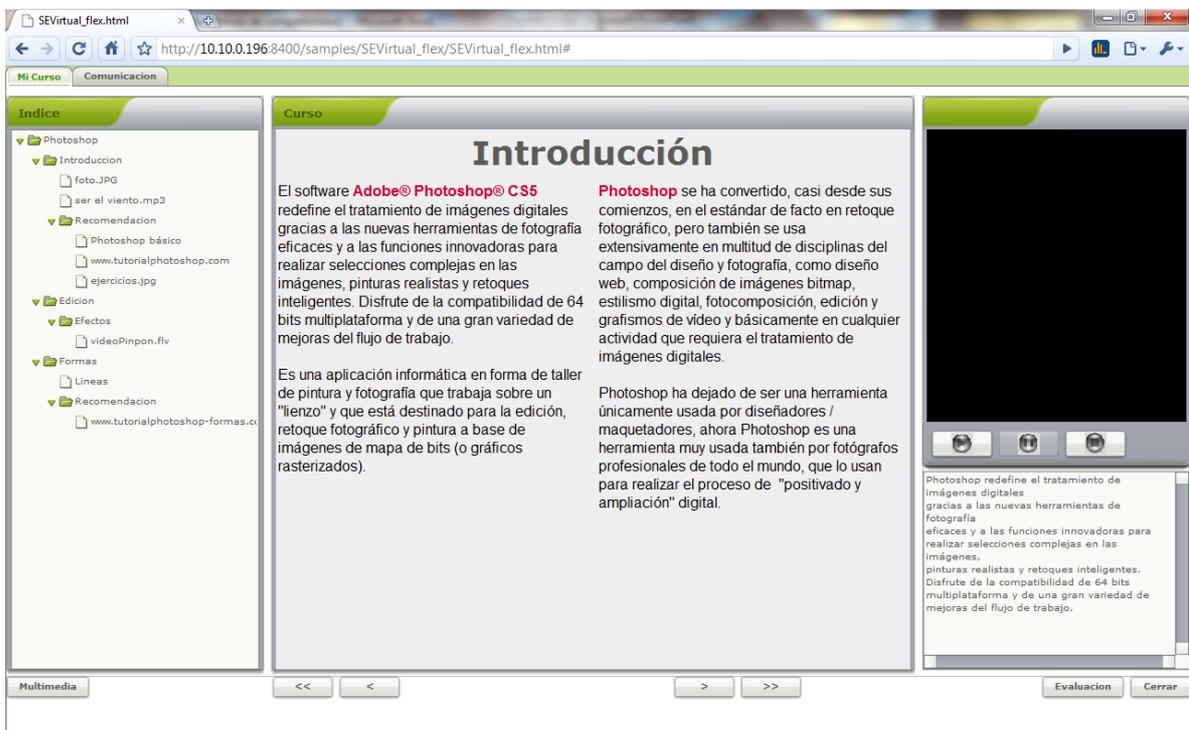


Figura 5.8 Curso Photoshop - primer alumno

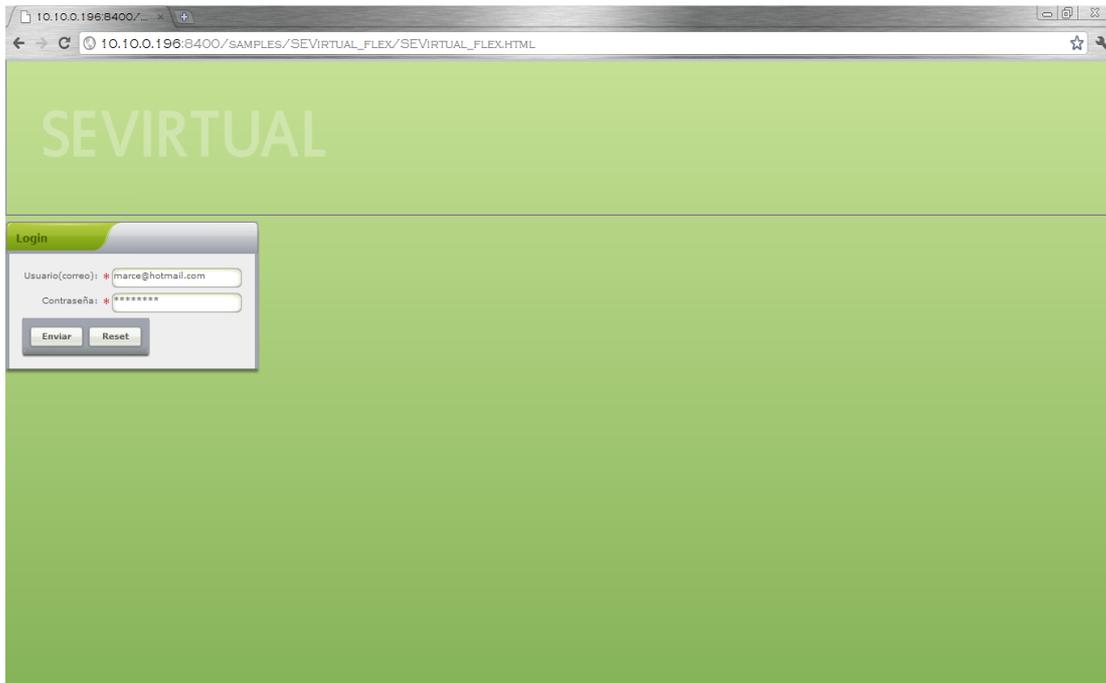


Figura 5.10 Login - segundo alumno

El segundo alumno visualiza el curso “Photoshop”, la **Figura 5.11** presenta cómo el alumno navega a través de la barra de navegación que se encuentra en la parte inferior de la pantalla, en este caso presionó el botón “<<” que lo regresa al inicio del curso, es decir, al tema “Introducción”.

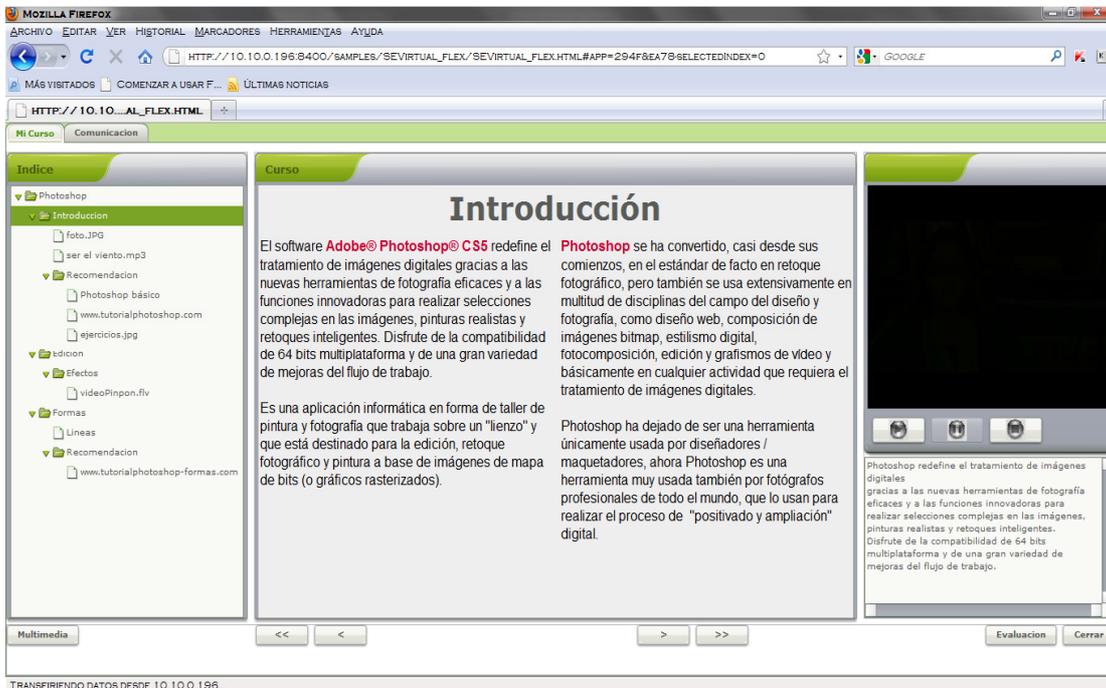


Figura 5.11 Curso Photoshop - segundo alumno

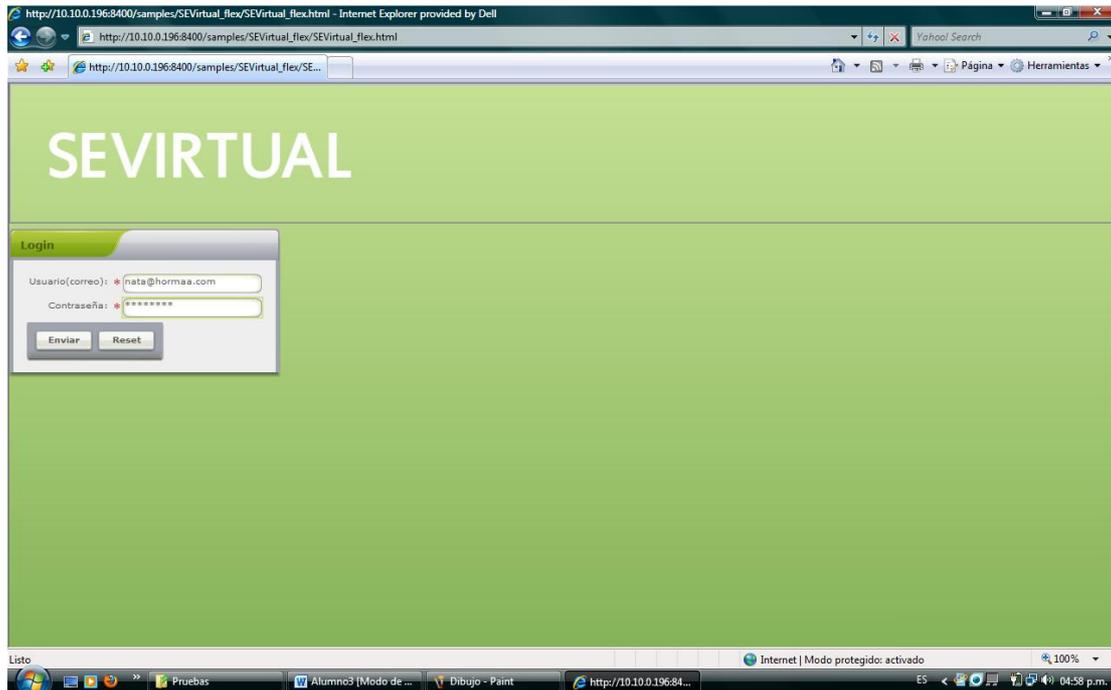


Figura 5.13 Login - tercer alumno

El tercer alumno visualiza el curso “Photoshop”, la **Figura 5.14** despliega cómo el alumno navega dentro del curso a través de la reproducción del video explicativo del curso, el cual automáticamente cambiará el tema dependiendo la reproducción del video.

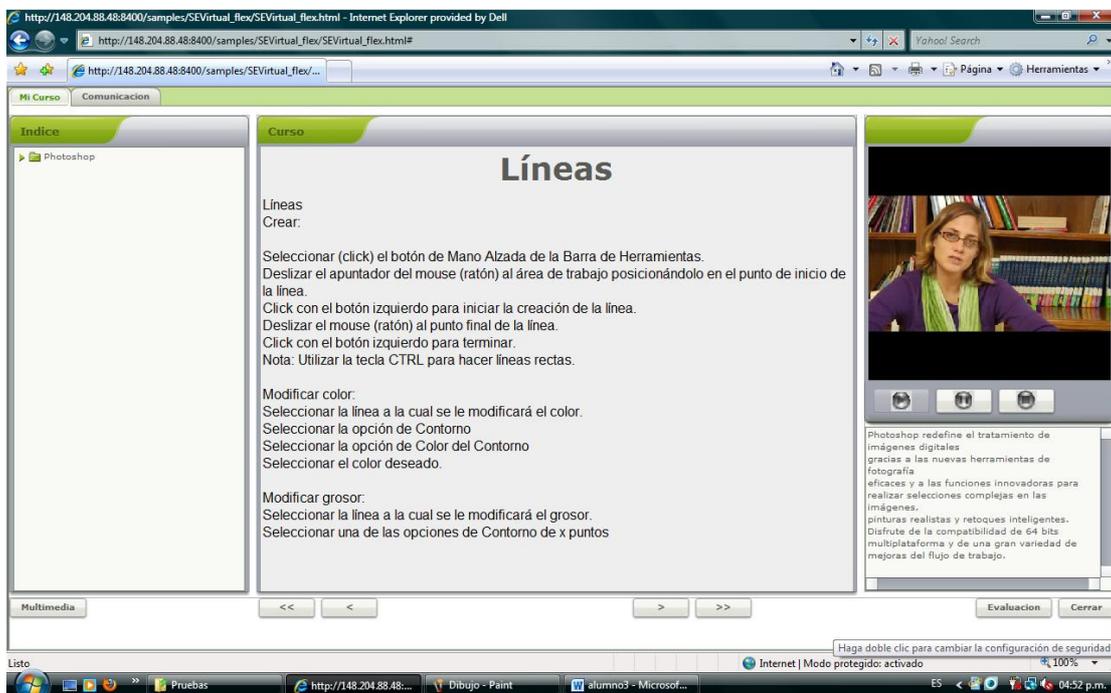


Figura 5.14 Curso: Photoshop - tercer alumno

Posteriormente el administrador modifica el nombre del usuario por: Moisés Almanza y lo guarda, la **Figura 5.17** muestra el resultado de haber actualizado al usuario.

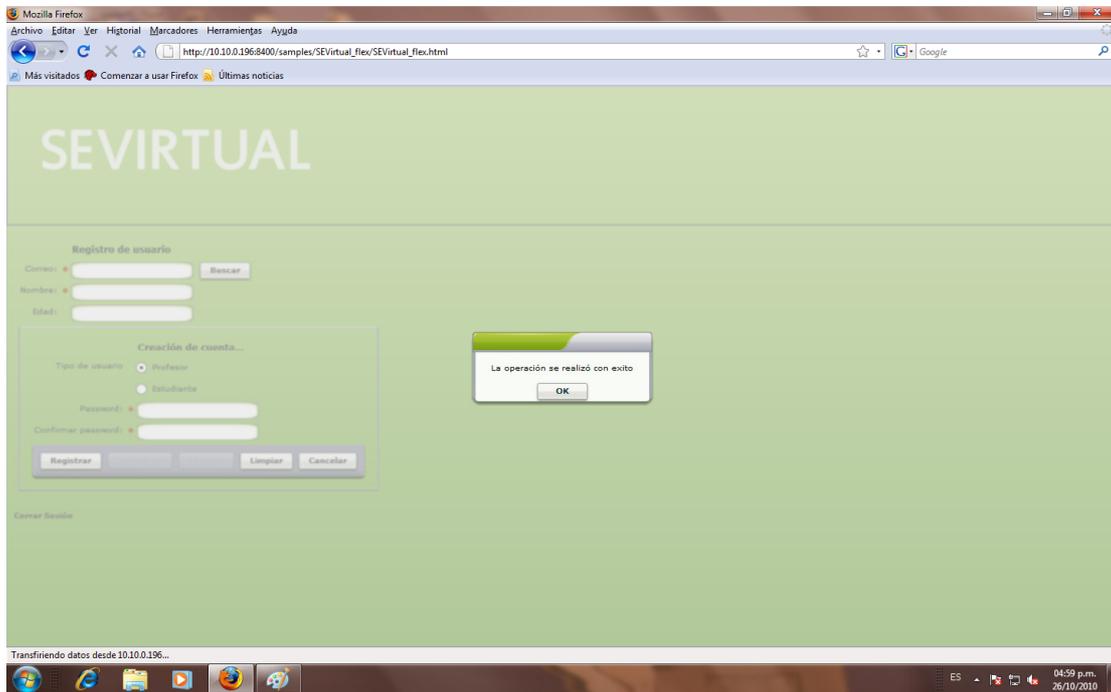


Figura 5.17 Actualización del usuario - administrador

El administrador busca al estudiante Moisés a través de su correo: alekine@hotmail.com para eliminarlo del sistema, la **Figura 5.18** presenta el resultado de la búsqueda con los datos actualizados y la **Figura 5.19** despliega el resultado de la eliminación del estudiante.

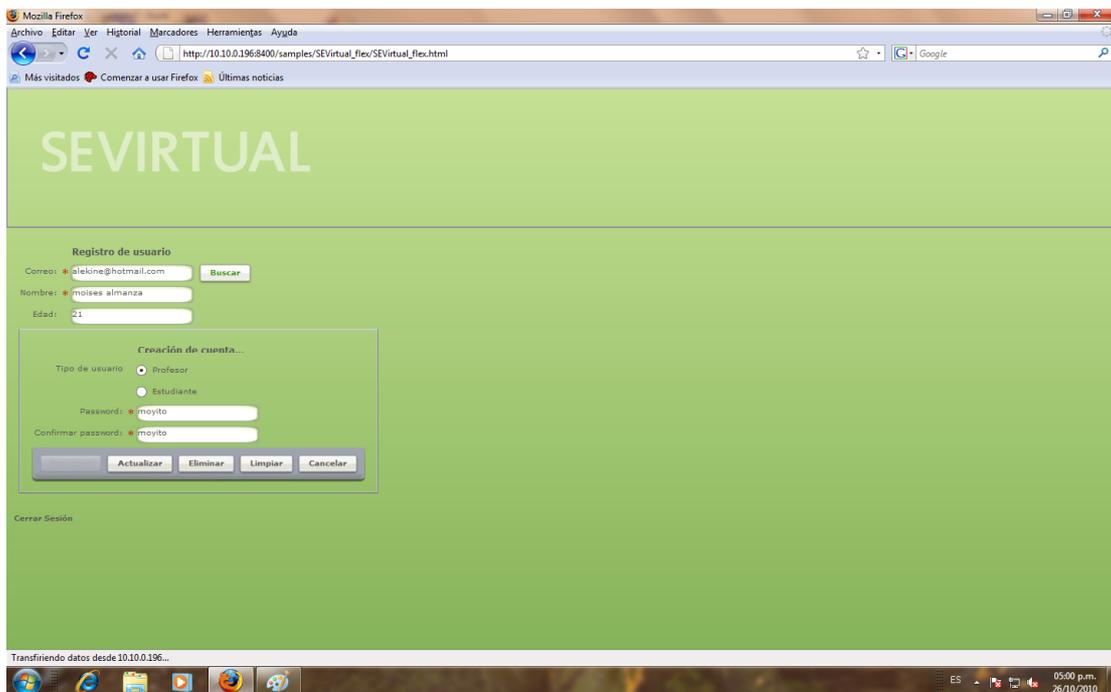


Figura 5.18 Búsqueda del estudiante modificado - administrador

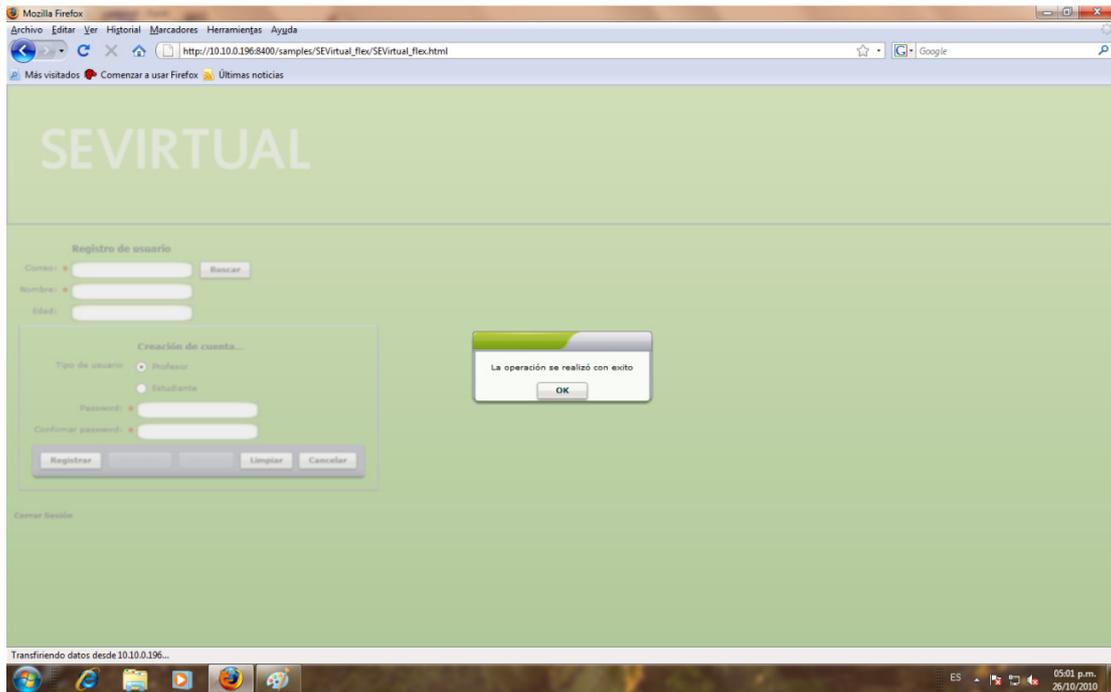


Figura 5.19 Eliminación del estudiante - administrador

Por otro lado el profesor sube un multimedia para el tema “Introducción”, la **Figura 5.20** muestra cómo el profesor ha subido la imagen foto.jpg al tema “Introducción”, la cual ya se ha almacenado en el servidor.

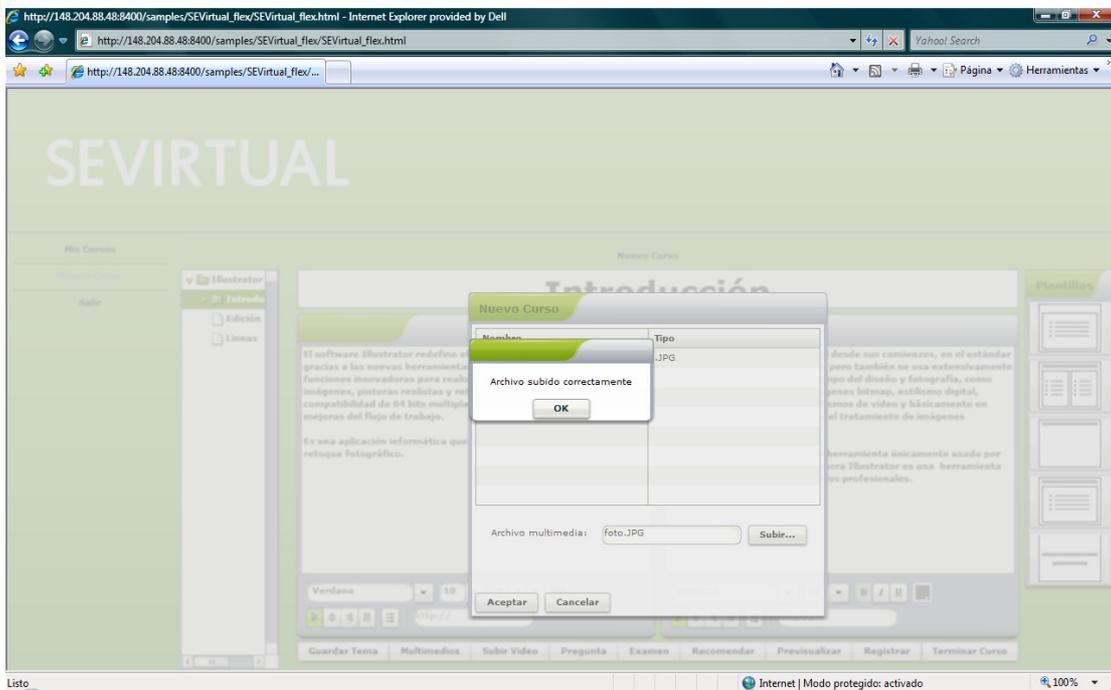


Figura 5.20 Multimedia agregado - profesor

El profesor sube el video explicativo del curso al dar clic en el botón "Subir video", la **Figura 5.21** despliega cómo el profesor ha subido el video: "photo.flv" y la **Figura 5.22** presenta cómo el profesor sube el texto que explica el contenido del video, en este caso sube el archivo: "texto.txt".

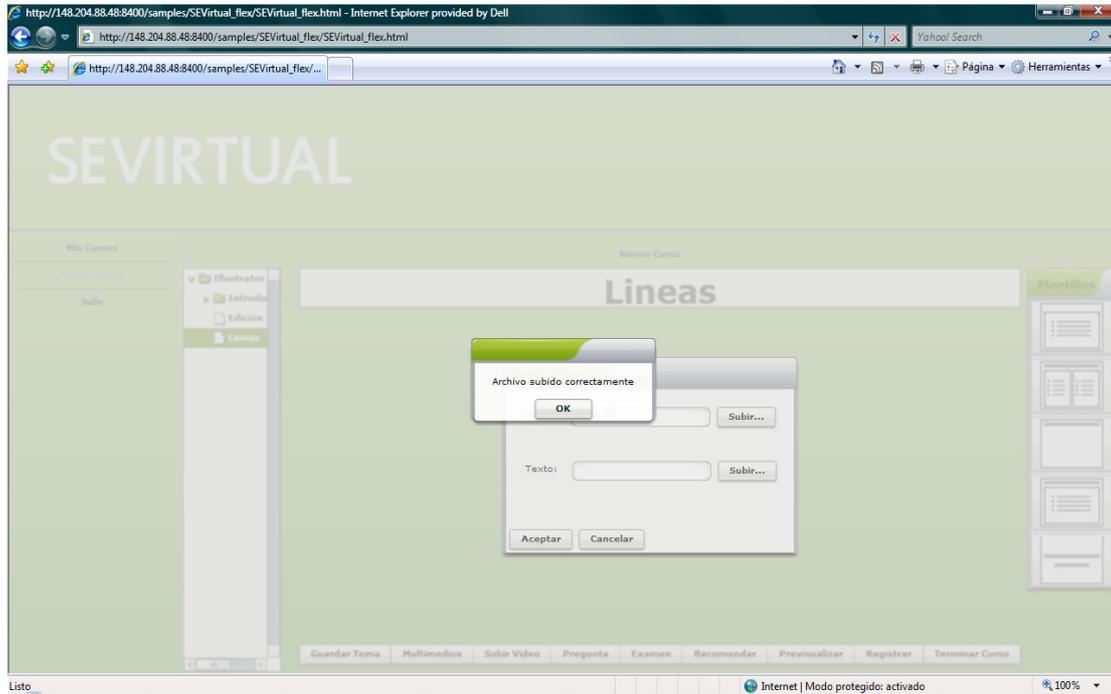


Figura 5.21 Video agregado – profesor

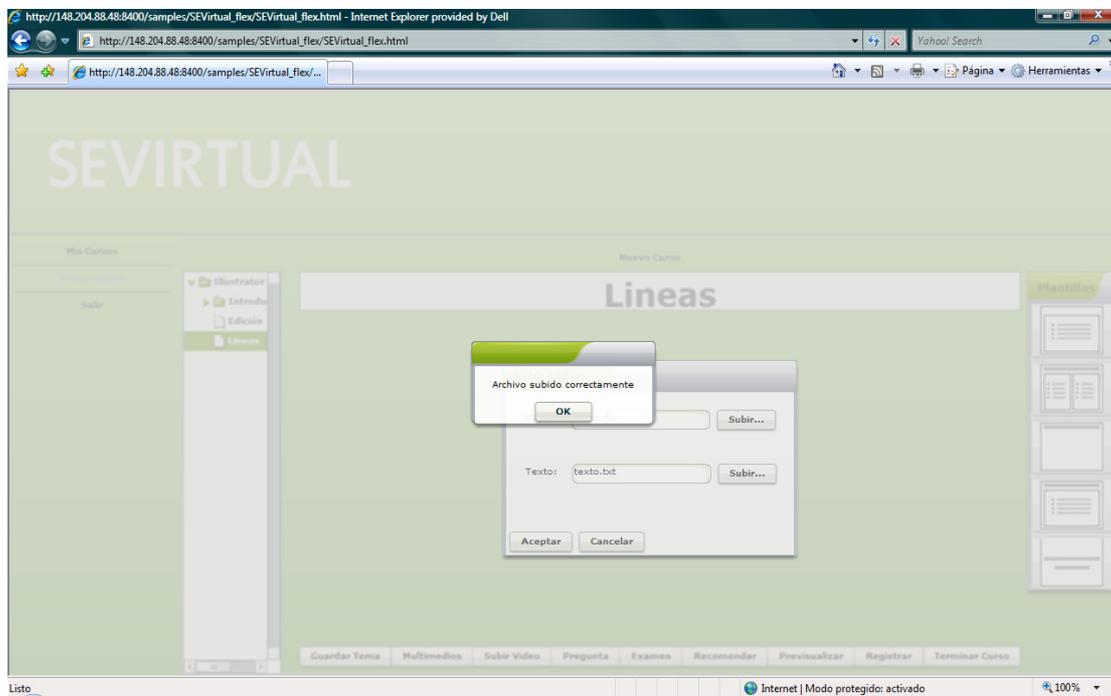


Figura 5.22 Explicación del video agregada – profesor

Ahora el profesor sincroniza la reproducción del video que subió al servidor con la lista de temas que ha creado, la **Figura 5.23** muestra cómo el profesor al dar clic en el botón "Cambiar tema" cambia automáticamente el tema en relación con la reproducción del video, internamente se crea un archivo de meta datos con la información de esta configuración.

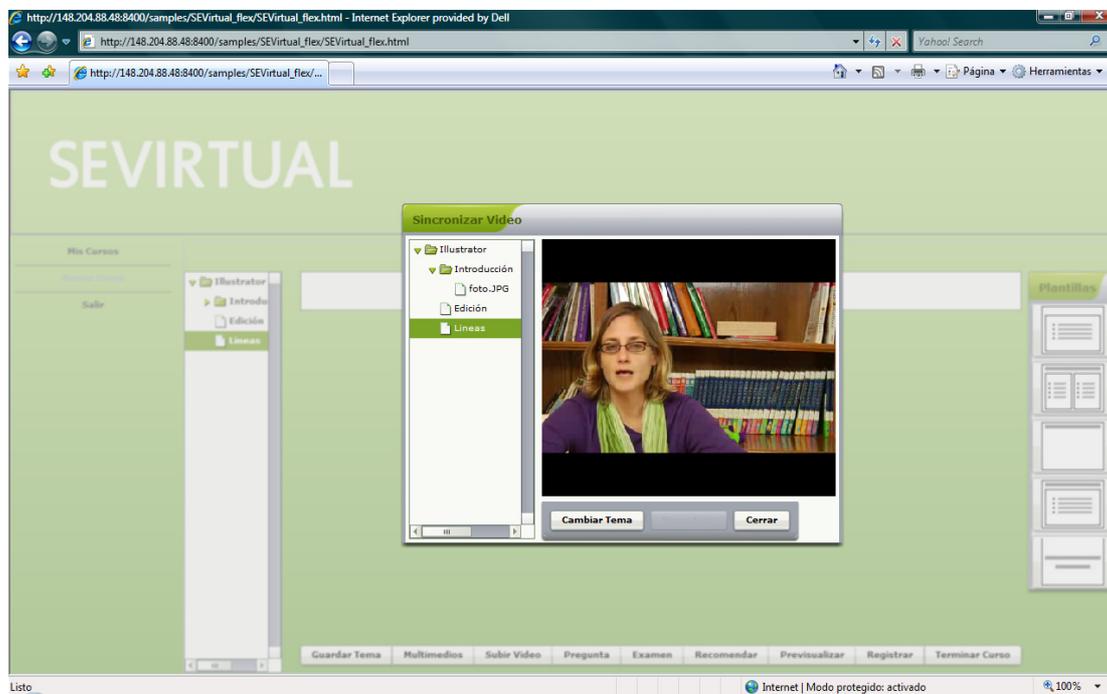


Figura 5.23 Sincronización del video con los temas – profesor

En otra instancia el primer alumno descarga un recurso del curso "Photoshop", la **Figura 5.24** despliega cómo el alumno descarga el archivo: "foto.jpg" y selecciona la ubicación para guardarlo.

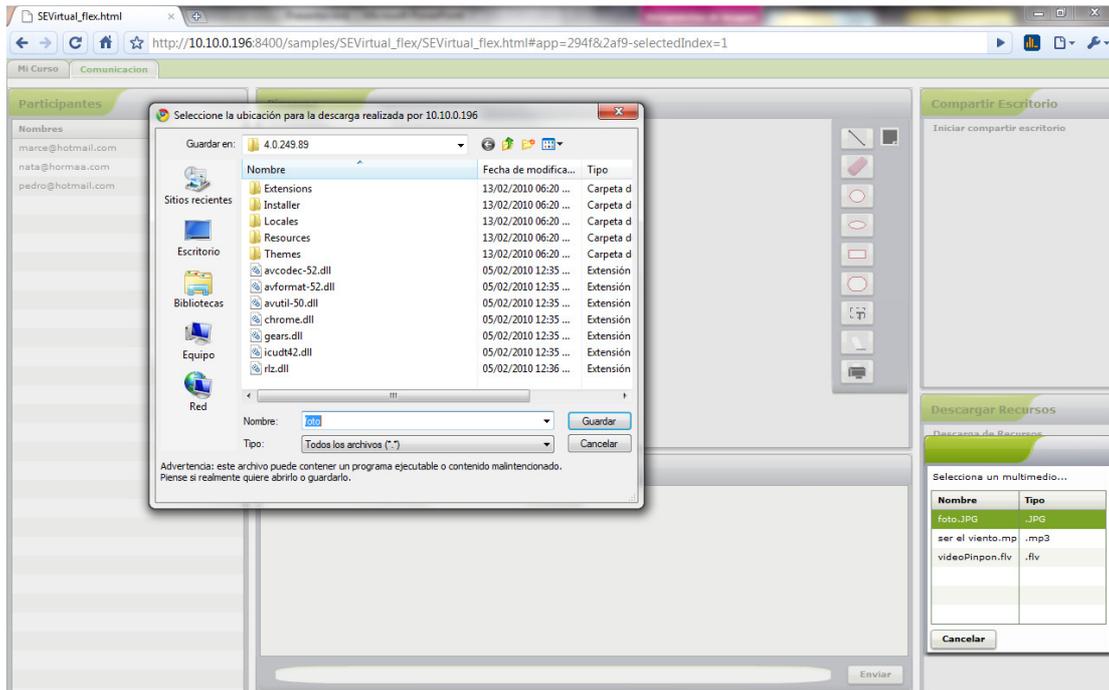


Figura 5.24 Descarga del recurso – primer alumno

El primer alumno empieza a interactuar con los participantes del curso al enviar un mensaje a través del chat, la **Figura 5.25** presenta cómo el estudiante envía el mensaje: “¿Qué nos permite hacer photoshop?”

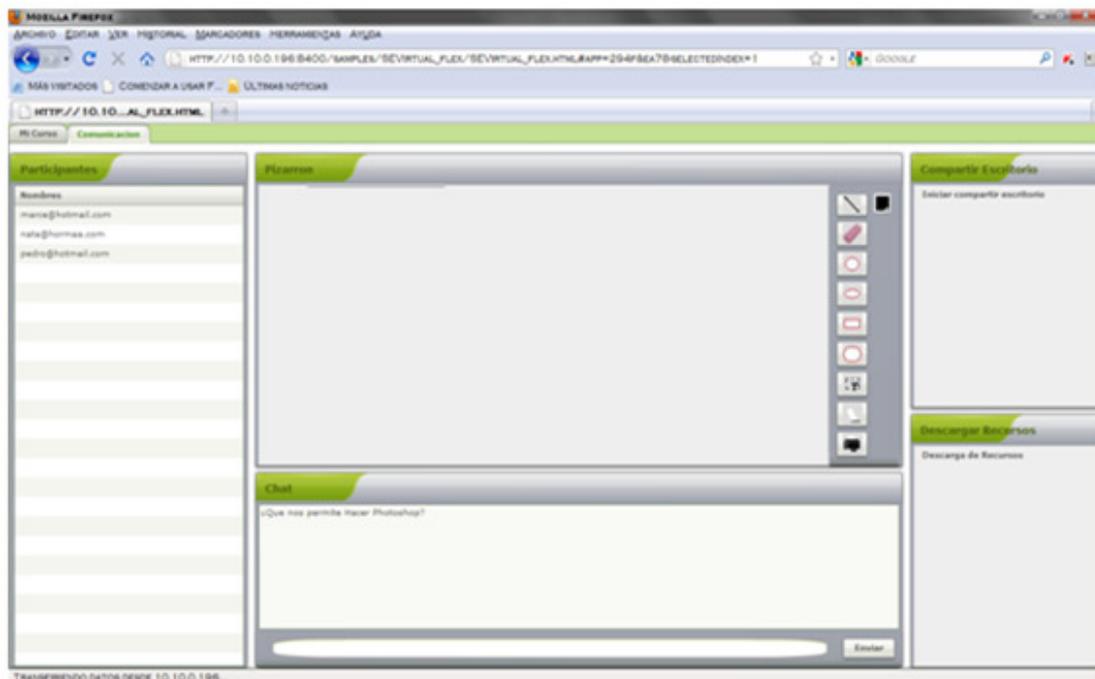


Figura 5.25 Uso del chat – primer alumno

El segundo alumno envía el siguiente mensaje: "Lo dibujaré en el pizarrón", respondiendo al primer alumno, la **Figura 5.26** muestra el mensaje enviado a los participantes del curso.

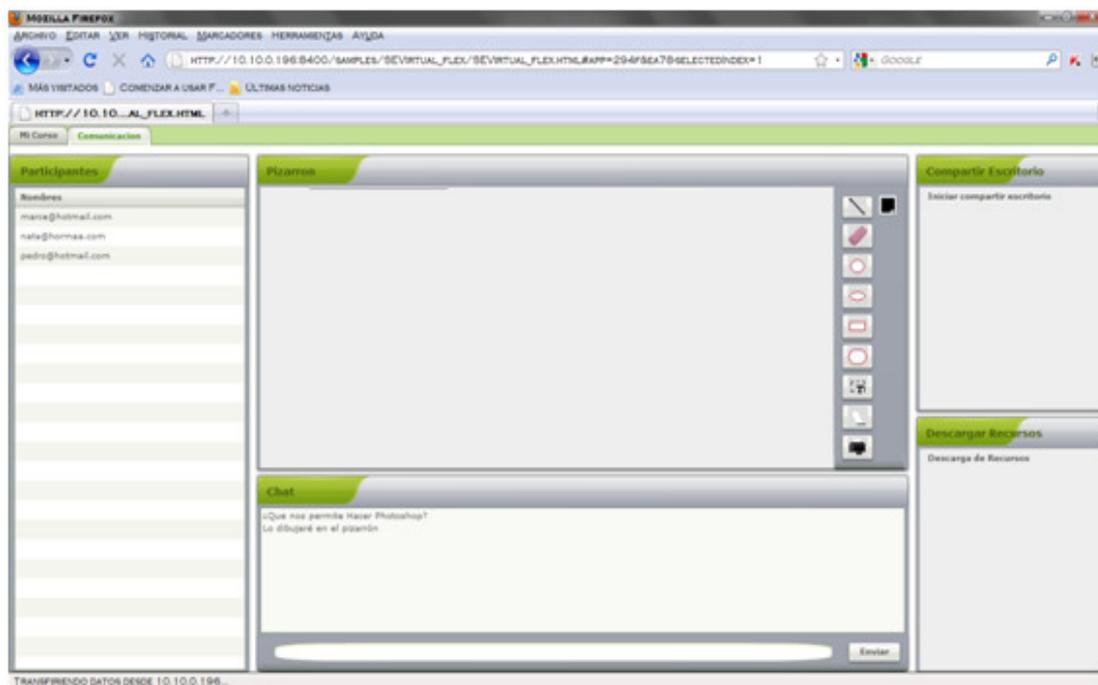


Figura 5.26 Uso del chat – segundo alumno

El segundo alumno comienza a dibujar a través del pizarrón un cuadro conceptual en el que le explica al primer alumno las funciones de Photoshop, la **Figura 5.27** despliega el cuadro conceptual enviado por el segundo alumno.

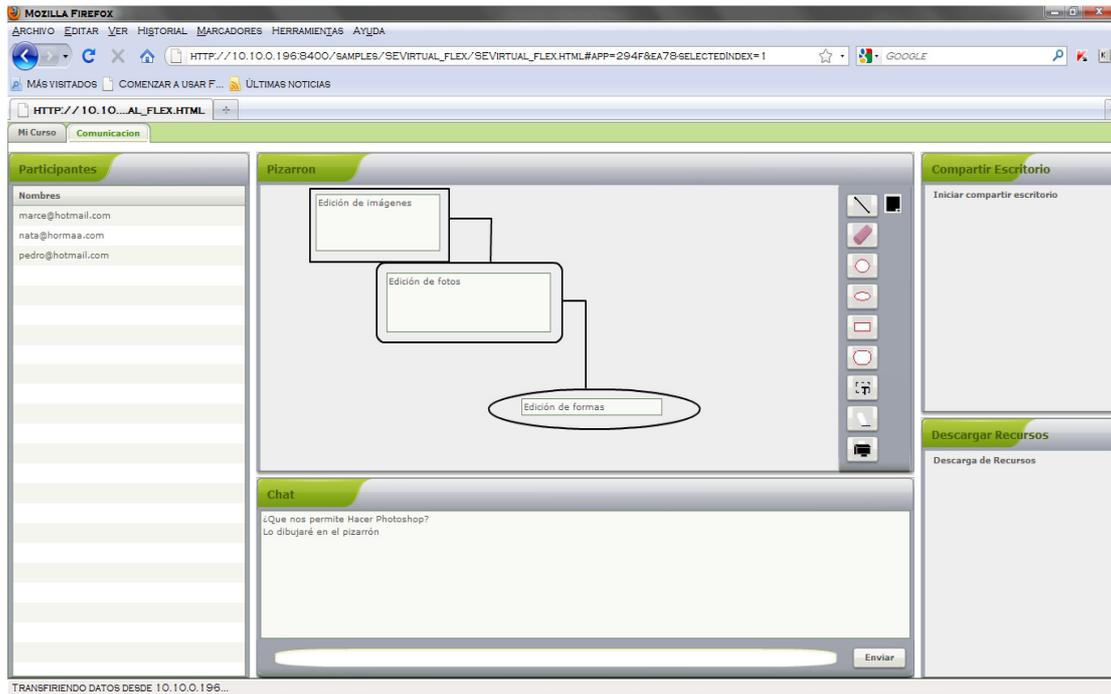


Figura 5.27 Uso del pizarrón – segundo alumno

Por otro lado el primer alumno envía el siguiente mensaje a los participantes del curso: “Voy a compartir el escritorio para enseñarte una duda que tengo de Photoshop”, la **Figura 5.28** muestra como el primer alumno empieza a compartir su escritorio al dar clic en la opción: “Iniciar compartir escritorio”.

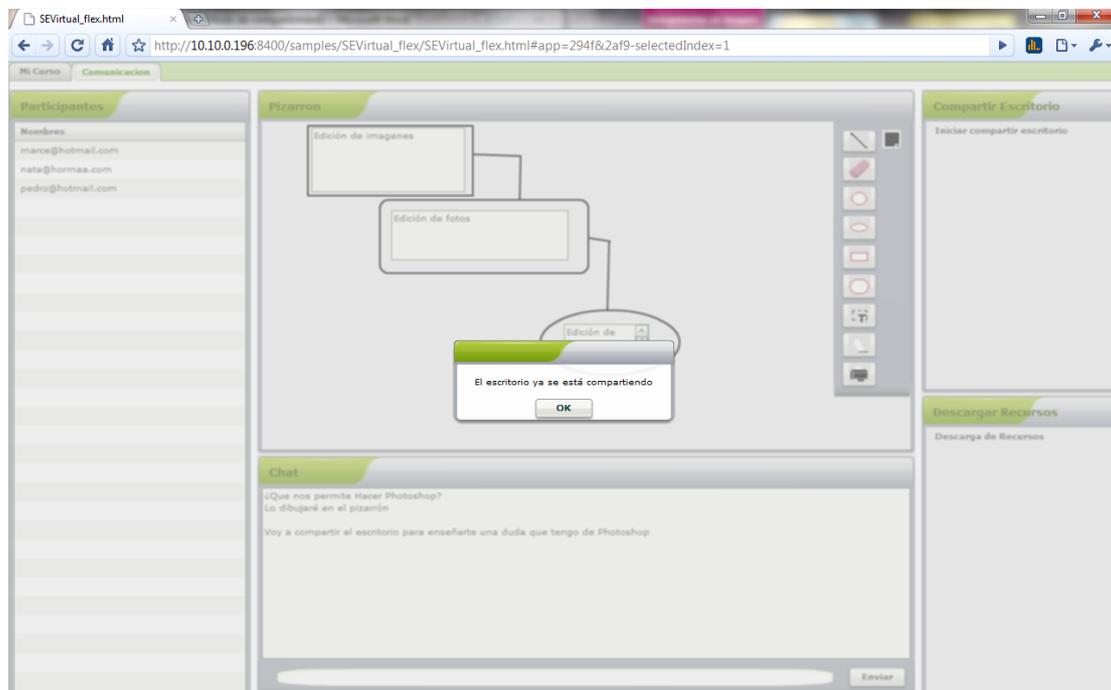


Figura 5.28 Compartición de escritorio – primer alumno

El segundo alumno visualiza el escritorio del primer alumno, la **Figura 5.29** presenta cómo el primer alumno comparte su escritorio y muestra a los participantes del curso sus aplicaciones.

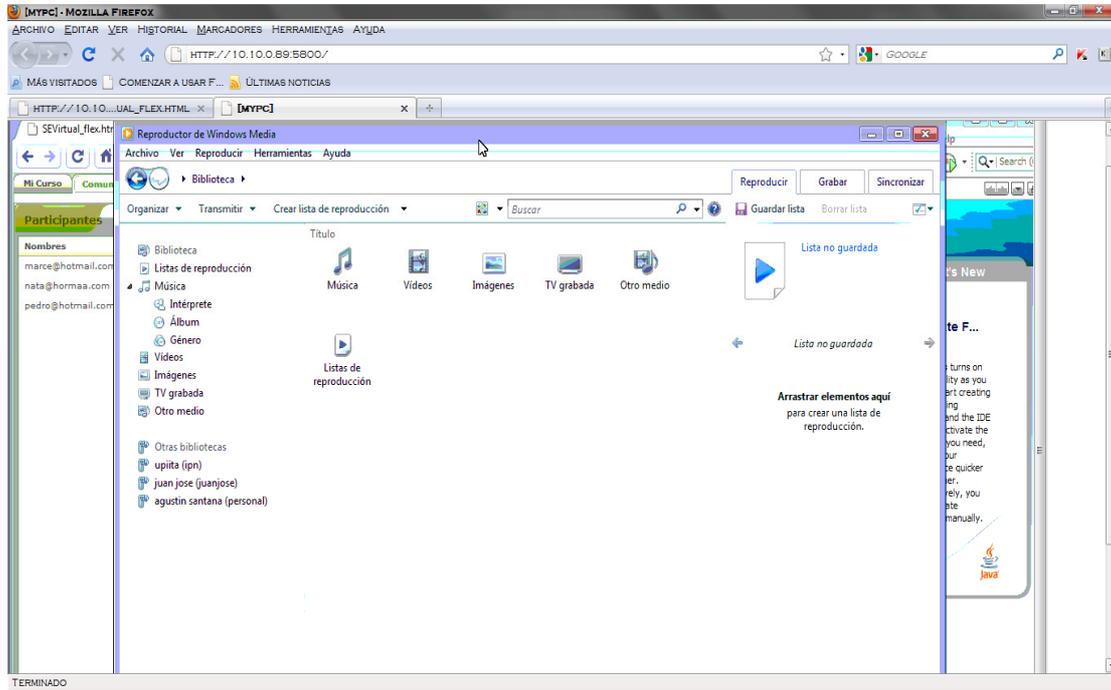


Figura 5.29 Visualización de escritorio – segundo alumno

En otra instancia el tercer alumno lleva a cabo la evaluación del curso al dar clic en el botón: “Evaluación”, la **Figura 5.30** muestra como el alumno responde una de las preguntas, en este caso una pregunta del tipo: “falso-verdadero”.

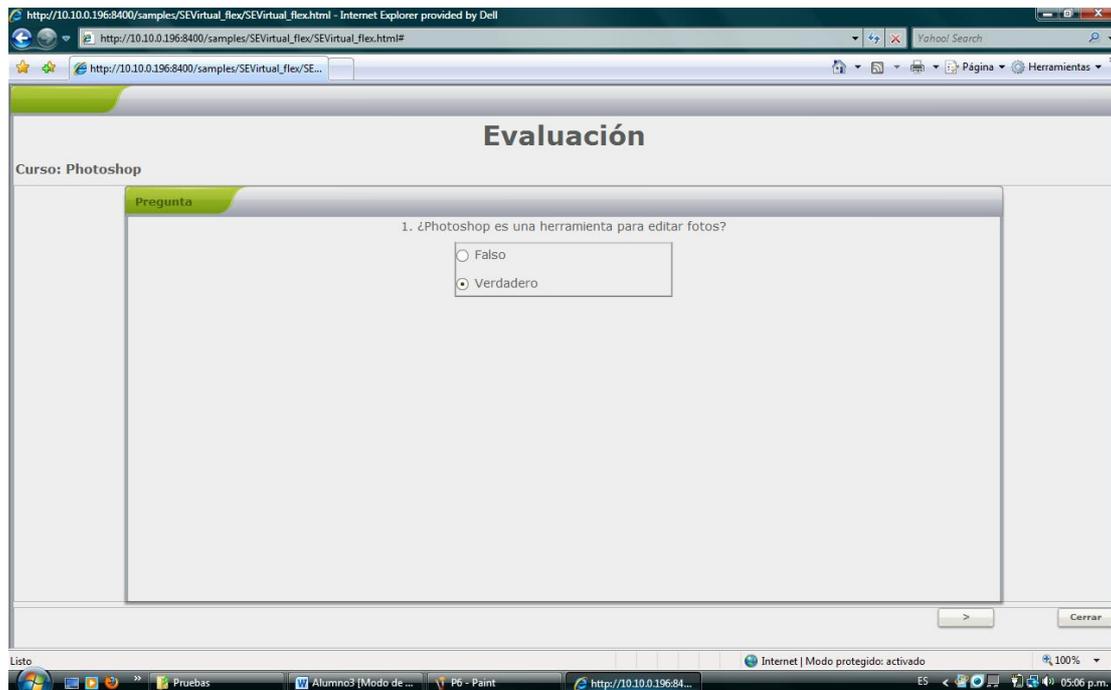


Figura 5.30 Evaluación – tercer alumno

Al terminar la evaluación, el profesor virtual a través del MAS le ofrece retroalimentación al alumno, la **Figura 5.31** despliega la retroalimentación que el tutor virtual ha dado al tercer alumno, el cual sólo se equivocó en dos preguntas relacionadas con el tema “Edición”, el tutor virtual proporciona la calificación obtenida por el alumno, le indica si el curso ha sido reconfigurado, las preguntas incorrectas, el tema relacionado con cada pregunta y a continuación le proporciona las recomendaciones, en este caso el tema edición no tiene ninguna recomendación agregada por el profesor, por lo tanto no se muestran.

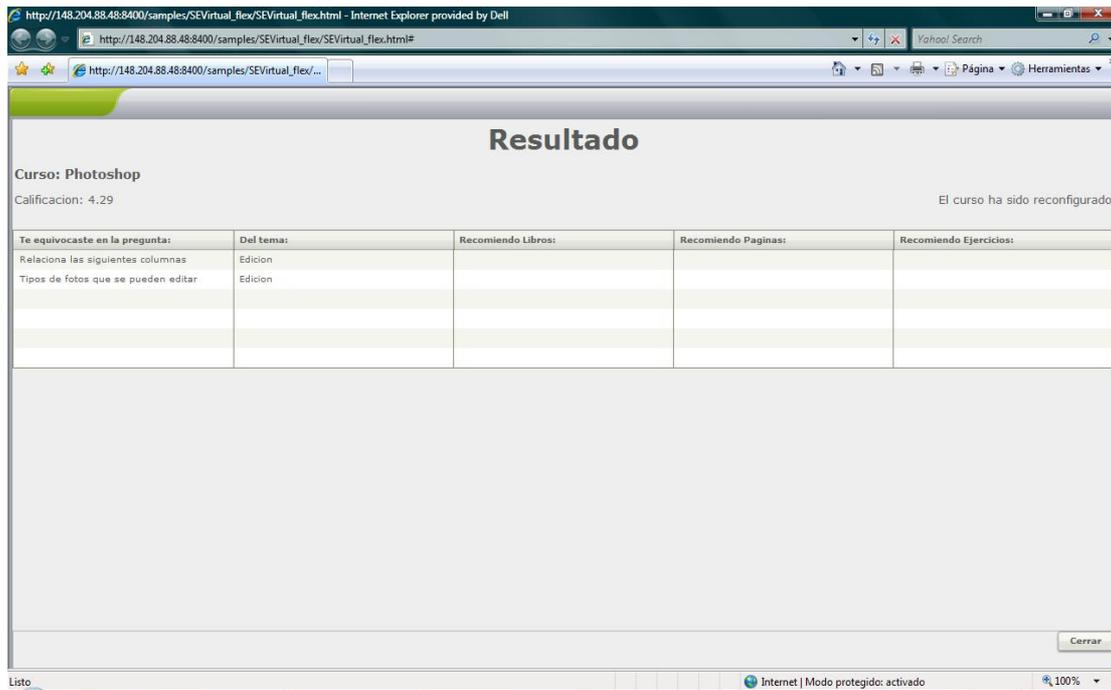


Figura 5.31 Retroalimentación – tercer alumno

El tercer alumno puede dar doble clic sobre alguna pregunta para recordar en qué pregunta se equivocó, la **Figura 5.32** muestra cómo el tercer alumno da doble clic sobre la pregunta “Relaciona las siguientes columnas” y a continuación el mismo componente de este tipo de pregunta se reutiliza para mostrar la pregunta.

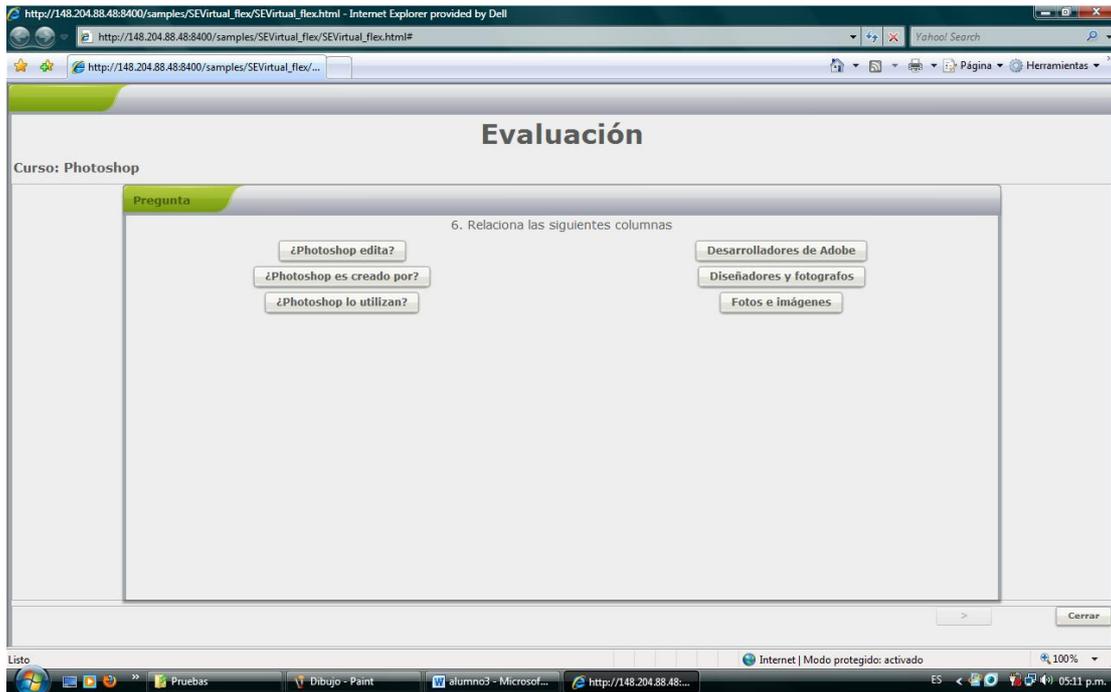


Figura 5.32 Pregunta incorrecta – tercer alumno

A continuación el tercer alumno desea ver el curso reconfigurado por lo que se sale del curso y da clic en la opción “Cursos reconfigurados”, la **Figura 5.33** despliega la lista de cursos reconfigurados, en este caso el tercer alumno solo tiene el curso que apenas le ha reconfigurado el profesor virtual.

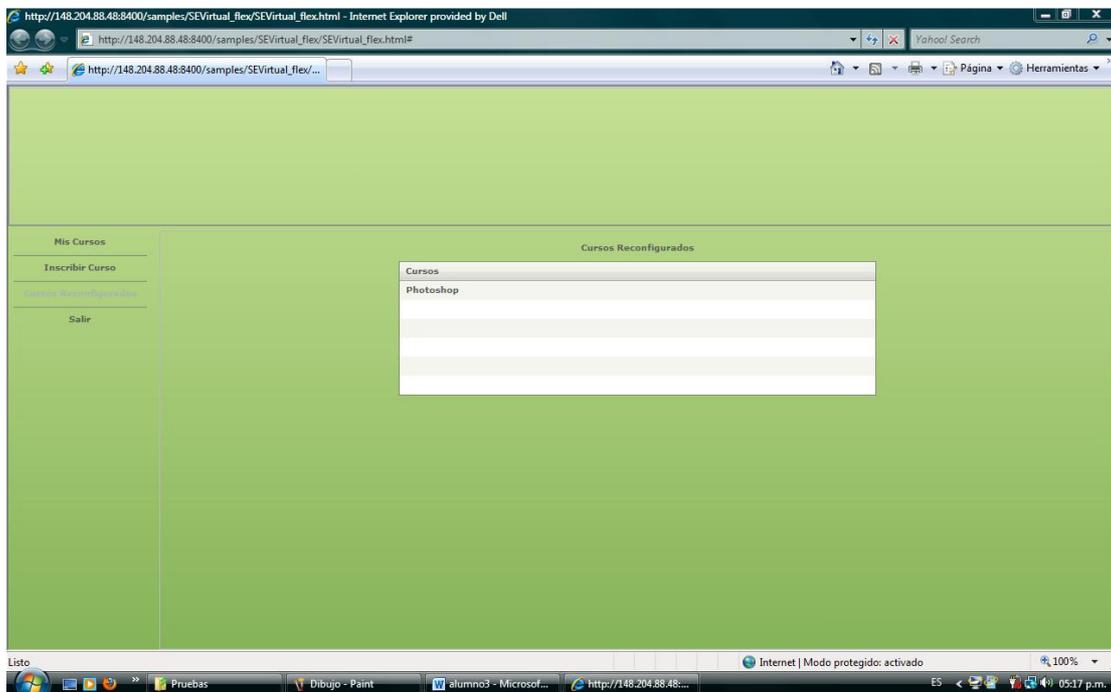


Figura 5.33 Cursos reconfigurados – tercer alumno

El tercer alumno da doble clic sobre el curso "Photoshop" para visualizarlo, el componente para visualizar cursos se reutiliza, la **Figura 5.34** despliega el curso "Photoshop" reconfigurado, el cual solo contiene un tema, el tema "Edición" que fue el tema que se le dificultó al alumno al responder las preguntas de la evaluación.

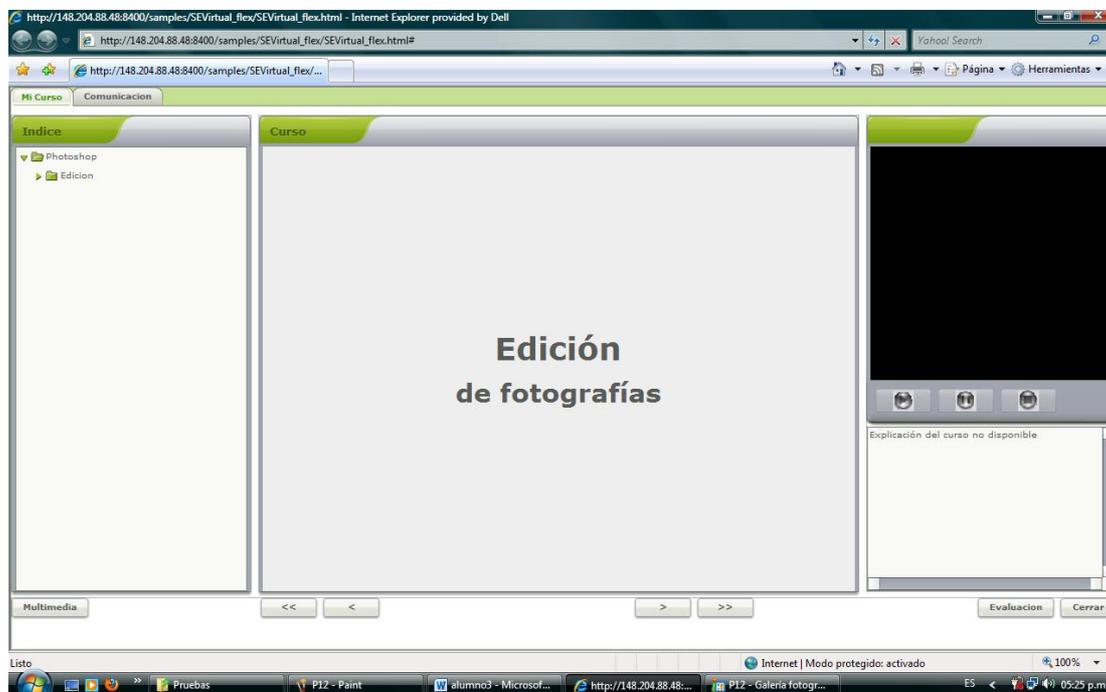


Figura 5.34 Curso: "Photoshop" reconfigurado– tercer alumno

En otra instancia, el profesor crea una pregunta del tipo: falso-verdadero, con los siguientes datos:

- Título: ¿Illustrator se utiliza para la edición de fotografías?
- Respuesta: verdadero.

La **Figura 5.35** muestra cómo el profesor agrega la pregunta de tipo: falso-verdadero con los datos anteriores y la **Figura 5.36** presenta el resultado de guardar la pregunta en el servidor al dar clic en el botón: "Guardar Pregunta".

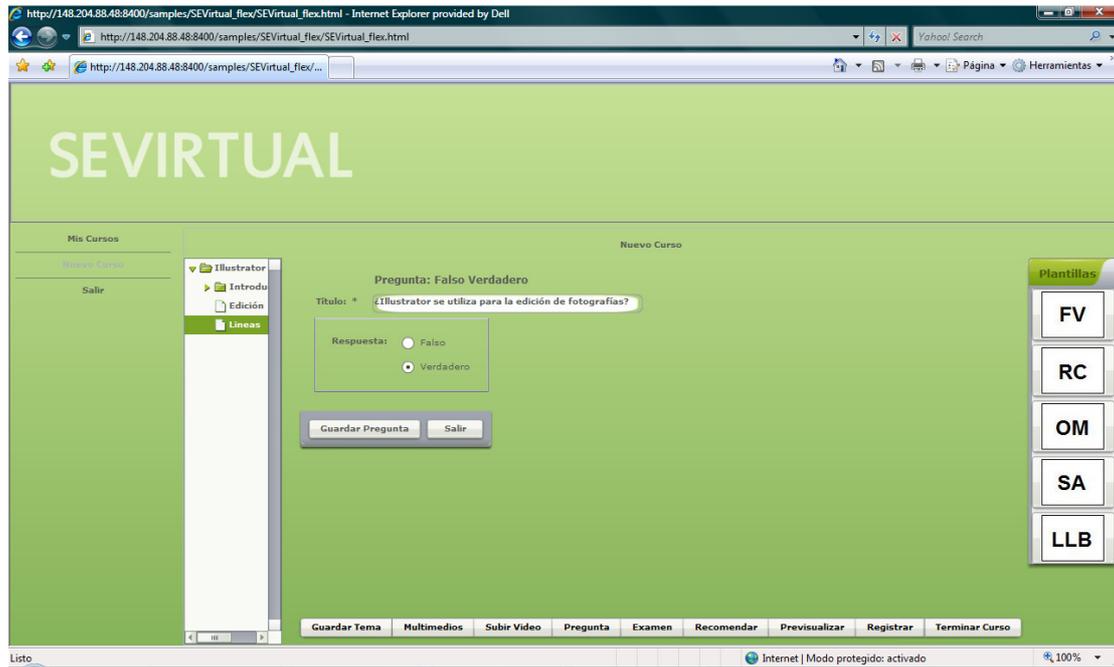


Figura 5.35 Curso: Pregunta: falso-verdadero – profesor

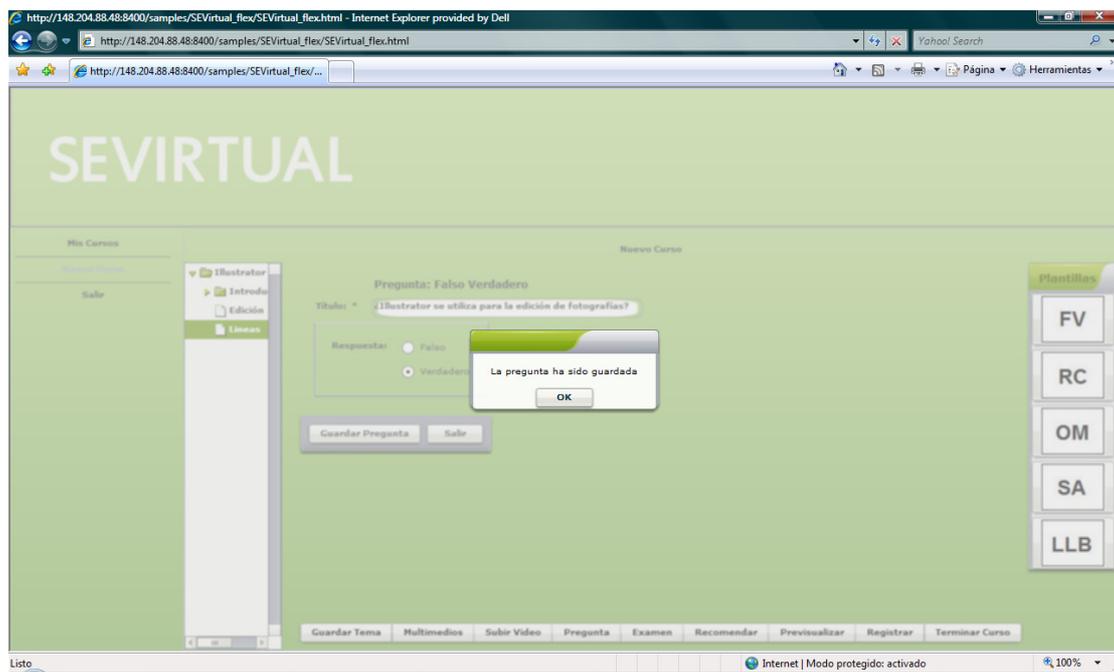


Figura 5.36 Pregunta guardada – profesor

El profesor crea un nuevo examen al dar clic en el botón: "Examen", la **Figura 5.37** despliega cómo el profesor guarda el nuevo examen en el servidor, al dar clic en el botón: "Guardar Examen".

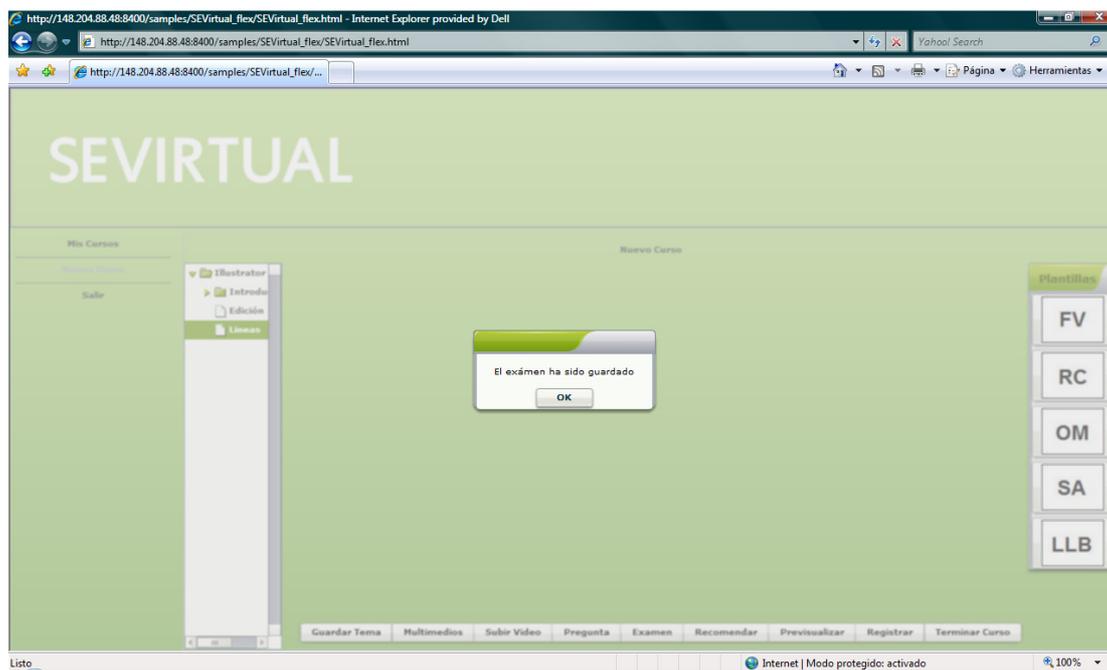


Figura 5.37 Examen guardado – profesor

5.2 RESULTADOS DE PRUEBAS DE SEVIRTUAL

Al realizar de manera concurrente las pruebas anteriores se obtuvieron los siguientes resultados:

- En general el sistema se comporta de manera estable.
- Debido a algunos problemas de conexión con la red experimentados en el laboratorio, los participantes tenían que volver a ejecutar la prueba.
- Se experimentó bajo rendimiento del sistema a la hora de cargar el video: "videoPinpon.flv", debido a la concurrencia y al estado de la red.
- Se experimentó bajo rendimiento del sistema a la hora de guardar la pregunta de tipo: falso-verdadero, debido a la concurrencia y al estado de la red.

5.3 COMPARATIVA ENTRE SISTEMAS DE ENTRENAMIENTO

En la **Tabla 5.1** se muestra un cuadro comparativo de las funcionalidades de SEVirtual con respecto a otros sistemas similares, debido a que el sistema ya fue desarrollado es posible comparar su funcionalidad con la de otros sistemas.

Los sistemas que se pudieron probar ya que fue posible el acceso a su funcionalidad fueron: Adobe Acrobat Connect Pro [27], eLecta Live [28] y Moodle [37], el Salón Virtual basado en *Web* [36] solo se comparó a través de la información publicada.

De las pruebas realizadas y de la información publicada se obtuvieron los resultados mostrados en la Tabla 5.1, cabe resaltar de estos resultados que SEVirtual permite la generación de los cursos de entrenamiento, por lo que el profesor no necesita tener ya creado el curso a través de presentaciones en Power Point o páginas HTML como en otros sistemas, permitiendo reducir la complejidad en cuando aspectos técnicos para la generación de cursos.

Por otro lado, los costos de licenciamiento de SEVirtual se reducen significativamente debido a la integración de herramientas de software libre dentro del sistema, mientras que en otros sistemas como Adobe Acrobat los costos de licenciamiento son muy altos como se mencionó en el capítulo 2, además el uso de componentes y meta datos permite la configuración de los componentes en tiempo de ejecución heredando las ventajas de los componentes como son: maximizar la reusabilidad, el control de cambios y reducir la complejidad.

A diferencia de los demás sistemas, SEVirtual simula a un profesor virtual a través de un MAS, que comienza sus funciones una vez que el estudiante es evaluado. El tutor virtual como ya se ha mencionado, brinda al estudiante retroalimentación dinámica, reconfiguración del curso en base a los temas que se le dificulten y proporciona recomendaciones en cuanto a libros, ejercicios y páginas de Internet sobre dichos temas.

Tabla 5.1 Comparativo de Sistemas de entrenamiento basados en *Web*

Características	Adobe Acrobat Connect Pro	eLecta Live	Salón Virtual basado en <i>Web</i>	Moodle	SEVirtual
Voz sobre IP	Si	Si	No	No	No
Audio y transmisión de video	Si	Si	No	No	Si
Compartir contenidos	Si	Si	No	Si	Si
Presentaciones Power Point	Si	Si	No	Si	El profesor no tiene que tener creado el curso, el sistema automatizará su generación.
Generación del curso (temas y contenido)	No	No	Si	No	Si
Compartir escritorios y ventanas	Si	Si	No	No	Si
Pizarrón	Si	Si	No	No	Si
Chat	Si	Si	No, usan correo electrónico	Si	Si
Programación basada en componentes	No	No	No	No se menciona	Si
Evaluación del alumno	No	No	Si	Si	Si
Costo de licenciamiento	Altos	Altos	No se menciona	No	Muy bajos
Compartición de sitio	No	Si	No	No	No

Web durante el curso					
Sistema Inteligente multiagentes	No	No	Si, para la parte de generación del curso	No	Si, para la parte de evaluación, retroalimentación, recomendación y reconfiguración del curso.
Profesor Virtual	No	No	Si	No	Si, en la parte del sistema asistencial basado en agentes.
Manejo de estándares internacionales	HTML	HTML	HTML	HTML	HTML, XML y basado en SCORM.
Manejo de calendario	Si	No	No	Si	No
Envío de recordatorio o invitación por correo	Si	No	No	Si	No
Generación de reportes y estadísticas	Si	No	No	Si	No, se considera como trabajo futuro
Grabación de sesiones	No	Si	No	No	No
Reconfiguración de curso, retroalimentación dinámica y recomendaciones	No	No	No	Retroalimentación por el profesor real	Si
Administración de tareas	No	No	No	Si	No
Administración de foros	No	No	No	Si	No

De las ventajas más importantes que tiene SEVirtual con respecto a los sistemas mencionados en la tabla anterior, se consideran: la automatización de la generación de los cursos de entrenamiento permitiendo que el profesor no sea un experto en aspectos técnicos solo necesita tener a la mano sus materiales y contenidos, mientras que en los otros sistemas el profesor ya necesita tener creado el curso; las herramientas colaborativas como: el pizarrón, el chat y la compartición de escritorio y el profesor virtual que a través de la evaluación del estudiante, permite la reconfiguración, recomendación y retroalimentación del curso.

5.4 RESUMEN

En el presente capítulo se mostraron los resultados de las pruebas realizadas a SEVirtual en las sesiones de cada usuario del sistema: el administrador, el profesor y el alumno. Mostrando para cada uno de todas las funcionalidades que tienen permitidas.

Por último se presentó la Tabla 5.1 comparativa con las funcionalidades de varios sistemas de entrenamiento comparándolos respecto a SEVirtual.

CAPÍTULO 6

CONCLUSIONES

En el presente capítulo se detallan las conclusiones del desarrollo de SEVirtual, entre ellas: los logros alcanzados, algunas aportaciones importantes como las publicaciones realizadas y el trabajo relacionado con este proyecto que se podría llevar en un futuro.

6.1 Logros alcanzados

El SEVirtual es una herramienta basada en el paradigma de educación basada en *Web*, que permite desarrollar materiales educativos para entrenamiento virtual como material didáctico de apoyo para el profesor en diferentes modelos educativos, el sistema apoya al profesor reduciendo la complejidad de elaboración de los mismos y, posibilitando la automatización de los materiales educativos para entrenamiento virtual, reduciendo costos, y produciendo materiales tipo RIA basados en el patrón de composición y la arquitectura MVC, proporcionando mantenibilidad y reusabilidad al sistema, además es una herramienta que proporciona asistencia al alumno en el proceso de enseñanza/aprendizaje por medio del sistema multi-agente desarrollado, posibilitando reconfiguración y secuenciación dinámica de los materiales. Además el uso de los metadatos dentro del sistema permitió maximizar la reusabilidad de los materiales educativos, posibilitó además la separación de contenido/navegación tipo ADL-SCORM, sincronización del curso y reducción de la base de datos, posibilitando al sistema flexibilizar su crecimiento basado en componentes de software. El ambiente es altamente interactivo lo que motiva al estudiante al permitir la colaboración entre los estudiantes, profesor y el MAS del sistema integrando un ambiente rico en experiencias enriquecedoras en su proceso de enseñanza/aprendizaje.

De acuerdo a los objetivos planteados en el capítulo uno y a los resultados obtenidos en el capítulo cinco se concluye lo siguiente:

El desarrollo del módulo de inicio de sesión permitió la autenticación y validación de los usuarios dentro del sistema a través de su cuenta (correo electrónico) y su contraseña.

El desarrollo del módulo del administrador facilitó al administrador el dar de alta usuarios tanto profesores como alumnos, modificar sus registros y eliminarlos.

El desarrollo del módulo del profesor permitió la generación automatizada de cursos de entrenamiento tipo RIA, en el que el profesor puede agregar y editar sus cursos, para que una vez listos sean registrados y el alumno tenga acceso a ellos.

El desarrollo del módulo del estudiante facilitó al estudiante la visualización de los cursos inscritos así como la evaluación del mismo a través de una serie de preguntas del tipo falso-verdadero, relación de columnas, opción múltiple, llenado en blanco y selección de área.

Estructurar los contenidos del curso en una estructura abierta a través de meta datos, permitió separar el contenido del curso de la parte de navegación en una forma similar a ADL-SCORM, maximizando la reusabilidad de los materiales educativos.

La arquitectura de meta datos desarrollada para la estructuración del curso y los contenidos, también es parte de la estructura de preguntas, evaluaciones y la sincronización del curso, dicha arquitectura facilitó la configuración de los componentes así como la reducción de la complejidad de la base de datos.

El desarrollo e integración de componentes colaborativos como son el chat, el pizarrón, el escritorio remoto, etc. posibilitan al alumno colaborar e interactuar en un ambiente colaborativo rico de enseñanza/aprendizaje. El alumno puede colaborar y comunicarse no solo con otros alumnos que estén conectados al curso sino también con el profesor que esté conectado en ese momento, e interactuar con el MAS del sistema en el momento de la evaluación.

Los componentes de generación de contenido posibilitan al profesor la generación de cursos de entrenamiento, permitiendo que el profesor no sea un experto en los aspectos técnicos, más bien sea un experto en su materia al reducir la complejidad de elaboración de sus materiales en los aspectos técnicos, y permitirle concentrarse en los contenidos y en nuevas didácticas dependiendo de su modo educativo usado.

El envío de métricas durante la evaluación permitió a través de la interacción con el MAS, un beneficio para el alumno ya que el profesor virtual se encarga una vez terminada su evaluación, de reconfigurarle el curso en base a los temas que se le hayan dificultado, proporcionándole retroalimentación dinámica y recomendación de libros, páginas de Internet y ejercicios relacionados con el tema, que el profesor haya propuesto en el sistema.

La integración de tecnologías dentro del sistema como: **WBE**, **RIA**, algunos patrones de diseño de software (el patrón de composición y la arquitectura **MVC**), servidores importantes como es el caso de: BlazeDS, Apache Tomcat y UltraVNC, el uso de varios lenguajes y estándares como Flex, Struts 2, JADEX, Servlet, Hibernate, JSP, XML, MXML, etc. y MySQL como el manejador de base de datos, permitió el desarrollo de los módulos anteriores, así como la comunicación con la aplicación del lado del servidor y la reducción significativa de costos de licenciamiento al ser herramientas de software libre.

El SEVirtual se basó en la arquitectura IEEE 1484 – LTSA, ADL - SCORM y el modelo **BDI** para el desarrollo del sistema multi-agente, lo cual permitió la simulación del profesor virtual.

El uso de la arquitectura **MVC** en el sistema permitió la reusabilidad y el mantenimiento del sistema, esta arquitectura fue implementada para el *middleware* del servidor haciendo uso del framework Struts 2.

El uso del patrón de composición en el sistema permitió maximizar la reusabilidad, vencer la complejidad y controlar los cambios modificando los componentes siempre y cuando se respete su interfaz, los componentes tanto indivisibles como compuestos ofrecen una interfaz única por medio de este patrón, la cual se debe de respetar para mantener compatibilidad hacia atrás con los componentes más viejos. Los componentes se reutilizan dentro del sistema en otros contextos cargando y descargando sus meta datos.

6.2 Aportaciones

Durante el desarrollo de este trabajo se realizaron dos artículos:

- Publicación y ponencia del artículo: "*Sistema de Entrenamiento Virtual basado en Web usando componentes de software*" en congreso internacional ANIEI [63].
- Artículo indexado para la revista: ESWA EXPERT SYSTEMS [64] (en elaboración).

En cuanto a las aportaciones del sistema:

Se diseñó una arquitectura de meta datos para la organización de la estructura del curso, los temas, las preguntas, las evaluaciones y la sincronización del video explicativo, esta arquitectura permitió separar la parte del contenido de la navegación de los cursos, además de maximizar la reusabilidad de los componentes, por otro lado el uso de esta arquitectura redujo la complejidad de la base de datos y también permite agregar en un futuro nuevos tipos de preguntas y temas, sin necesidad que modificar la estructura de la base de datos.

Se implementó una arquitectura basada en componentes que hereda todas las ventajas del patrón de composición como son:

- Maximizar la reusabilidad al cargar y descargar los meta datos que configuran a los componentes en tiempo de ejecución y al reutilizar los componentes en otros contextos dentro de la aplicación, por ejemplo el componente de visualizar curso se reutiliza tanto al previsualizar el curso en el módulo del profesor, como al visualizar el curso en el módulo del alumno.
- Reducir la complejidad, mediante el uso de componentes de software.
- Controlar los cambios actualizando los componentes y agregando nuevos componentes siempre y cuando se respete su interfaz.

Se desarrolló una arquitectura para el MAS haciendo uso del modelo BDI que permitió crear agentes de software proactivos orientados al logro de sus metas, para la reconfiguración, retroalimentación y recomendación.

Se implementó la tecnología de las aplicaciones RIA que son la evolución de las aplicaciones *Web* normales para proporcionar al usuario aplicaciones ricas en multimedios, más interactivas y con un canal de comunicación bidireccional asíncrono con el servidor mejorando el rendimiento del sistema, lo cual proporciona una experiencia más agradable al usuario.

Se redujeron significativamente los costos de licenciamiento al utilizar herramientas de software libres.

Por último, la integración de todo el sistema permitió la generación automatizada de cursos de entrenamiento ya que muchas veces los cursos se generan de manera artesanal, uno por uno, y esto consume más tiempo y esfuerzo para el profesor.

6.3 Trabajo futuro

Como trabajo futuro se plantean mejorar varios aspectos del sistema SEVirtual, entre los que destacan los siguientes: validación de cambios que realice el profesor al curso y su salvado en la estructura del curso, para su persistencia en el servidor y no estar dando clic en el botón "Guardar Tema", también es necesario trabajar en el sistema la posibilidad de agregar recursos adicionales tipo PDF, DOC, etc., para que los estudiantes los puedan descargar e incluso algunos ejercicios podrían venir en esos formatos.

Otro punto a considerar es la eliminación de archivos en el servidor al borrar algún curso de entrenamiento del sistema. En la parte de evaluación sería útil agregar algún tipo de multimedia a las preguntas.

Otro aspecto a trabajar es la generación de reportes, estadísticas de las evaluaciones y el comportamiento del estudiante durante el curso para que el profesor cuente con información útil del estudiante, para una mejor toma de decisiones respecto al estudiante. Es importante un módulo de estadísticas gráficas para que el profesor pueda identificar los temas de mayor dificultad del curso, permitiendo al profesor identificar problemas en cuanto a sus contenidos con los estudiantes, teniendo la posibilidad de revisarlos y en caso necesario adecuarlos a las necesidades de los estudiantes.

Otro módulo a considerar en el sistema es el de calendarización de actividades del curso y notificaciones a los alumnos vía correo electrónico.

Otro aspecto a mejorar es el envío de más métricas al MAS con la finalidad de personalizar adecuadamente las respuestas hacia los estudiantes, un ejemplo sería la trayectoria del alumno a lo largo del curso, así como su evaluación y los tiempos de respuesta a las preguntas, etc., con la finalidad de que el algoritmo de retroalimentación tenga más elementos para personalizar su respuesta al estudiante.

Adicionalmente, es posible mejorar la arquitectura del sistema multi-agente y agregar más funcionalidades al profesor virtual e incluso integrar ciertos algoritmos de inteligencia artificial.

Bibliografía

- [1] Don Morrison, (2003) *E-Learning Strategies: How to get implementation and delivery right first time*. WILEY.
- [4] Shirley Alexander, (2001) "E-learning developments and experiences", *Education + Training*, Vol. 43 Iss: 4/5, pp.240 - 248.
- [5] Ruimin Zhang, Bofeng Zhang, Jingchen Zhu, Huiting Huang, (2008) *Development of Multi-Video Based Virtual Classroom and its Application in English as Second Language Learning*, IEEE Computer Society. Shanghai, China. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4731401&isnumber=4731352>
- [6] Colin Steed, (1999) *Web-based Training*, WILEY.
- [7] Reva Freeman (2000), *What is an Intelligent Tutoring System?*, Intelligence. Northern Illinois University.
- [8] Juan Humberto Sossa, Herbert Freeman, Carlos Vizcaino Sahagún, (2001) *Arquitectura de componentes de software programables reutilizables inteligentes para materiales multimedia en EVA*, Congreso Internacional de Computación 2001 Memoria CIC-IPN, México, D. F.
- [9] Santos, J, (2005) *La universidad chilena desde los extramuros: Luis Scherz García*, Santiago: Universidad Alberto Hurtado.
- [10] Jacobo Moquete, (1995) *Introducción a la Educación*, Malibú y Editora Tavarez, Santo Domingo.
- [11] Harasim, L, (1990) *Online Education: An Environment for Collaboration and Intellectual Amplification*, Preaguer Pub., New York.
- [12] Butterfield J & Krishnamurthy R, (2000) *Virtual Summer School*, *TESOL Spain Newsletter*, Vol 23.
- [13] Sangrà, M. A., (2002) *Educación a distancia, educación presencial y usos de la tecnología: una triada para el progreso educativo*. Edutec. Revista Electrónica de Tecnología Educativa, No. 15, Universitat Oberta de Catalunya, España.
- [14] Jackson Bob, (2001) *E-Learning Quality: The Concord Model for Learning from a Distance*, NASSP Bulletin, Vol. 85, No. 628, pp. 37-46.
- [15] José Luis García Llamas, (1991) *La Universidad Nacional de Educación a Distancia. Aproximación a la evaluación de un modelo innovador*, Ministerio de Educación y Ciencia, No. 60.
- [16] Michael Moore, (2001) *La educación a distancia en los Estados Unidos: estado de la cuestión*, UOC.
- [19] George Magoulas, Sherry Chen, (2006) *Advances in Web-Based Education*, INFOSCI.
- [20] Thomas D. Lynch, Cynthia E. Lynch, (2000) *Web-Based Education*, *Innovation Journal*, Vol. 8, No. 4.
- [21] Anil Aggarwal, (2003) *Web-Based Education: Learning From Experience*. INFOSCI.

- [23] Wiley, D. A., (2000) *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*. URL: <http://reusability.org/read/chapters/wiley.doc> (consultado el 15 de noviembre de 2009).
- [24] L'Allier, J. J., (1997) *NETg's precision skilling: The linking of occupational skills descriptors to training interventions*. NETG.
- [29] Qingtang Liu, Chengling Zhao, Zongkai Yhan, (2003) *Construction of a Web-Based Virtual Classroom and its effective analysis*, IEEE Computer Society. Wu-Han, China.
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1264745&isnumber=28283>
- [32] Ángel García-Beltrán, Raquel Martínez, (2007) *Aula Web 2007*, Madrid.
- [34] Erdoza, (2008) *Introducción al Modelo ADL SCORM 1.*, Erdoza Consultores.
- [35] Advanced Distributed Learning, (2006) *SCORM 2004 3^d Edition Run-Time Environment*. Versión 1.
- [36] Vladimir Trajkovic, Danco Davcev, Goran Kimovski, Zaneta Petanceska, (2000) *Web – Based Virtual Classroom*, IEEE Computer Society. Skopje, Macedonia. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=868966&isnumber=18796>
- [41] Szyperski and Pfister, (1997) *Import is not inheritance – Why we need both: modules and classes*, European Conference on Object-Oriented Programming (ECOOP).
- [42] C.Szyperski, (1999) *Component Software -Beyond Object-Oriented Programming*, Addison-Wesley.
- [43] Nicholas R. Jennings, Michael J. Wooldridge, (1998) *Agent Technology*, Springer.
- [44] R. Peredo Valderrama, Leandro Balladares Ocaña, Alejandro Canales Cruz y Alain Menchaca R., (2008) *Arquitectura para Sistemas de EBW usando programación orientada a componentes*, Rev. Diálogo Educ., Curitiba, v. 8, n. 24, p. 485-502.
- [46] Eli Noam. (1995) *Electronics and the dim future of university*. NETG.
- [50] Michael Jesse Chonoles, James A. Schardt, (2003) *UML 2 for Dummies*, Wiley.
- [51] Craig Larman, (2003) *UML y Patrones. Introducción al análisis y diseño orientado a objetos*, Pearson.
- [53] Booch, Jacobson y Rumbaugh, (1997) *The UML specification documents*, Addison-Wesley, Santa Clara, CA.
- [62] Michael Bratman, (1987) *Intentions, plans, and practical reason*. Harvard University Press.
- [63] Adriana N. Ramírez Salazar, Rubén Peredo Valderrama, Alejandro Canales Cruz, (2010) *Sistema de Entrenamiento Virtual basado en Web usando componentes de software*, Congreso Internacional ANIEI, Puerto Vallarta, Jalisco.

Referencias Web

- [2] Milenium, URL: <http://www.informaticamilenium.com.mx/Paginas/mn/articulo78.htm> (consultado el 4 de agosto de 2009)
- [3] Master Magazine, URL: <http://www.mastermagazine.info/termino/3934.php> (consultado el 21 de agosto de 2009)
- [17] Proyectos de educación a distancia en Venezuela, URL: <http://neutron.ing.ucv.ve/revista-e/No5/MQuevedo.htm> (consultado el 11 de septiembre de 2009)
- [18] EduDigital, URL: http://www.edudigital.unellez.edu.ve/portal/index.php?option=com_content&task=view&id=26&Itemid=58 (consultado el 11 de septiembre de 2009)
- [22] Arquitectura Cliente/servidor, URL: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf (consultado el 11 de septiembre de 2009)
- [47] Laboratorio de Cibermetría, del Consejo Superior de Investigaciones Científicas de España (CSIC), URL: http://www.webometrics.info/top6000_es.asp (consultado el 15 de agosto de 2009)
- [48] Nua Internet Surveys, URL: <http://www.nua.ie/surveys> (consultado el 15 de agosto de 2009)
- [49] *** Datos de Júpiter Communications, Enero de 2000, URL: <http://www.jup.com> (consultado el 15 de agosto de 2009)
- [25] GestioPolis: ¿Entrenando?... Entonces serás el mejor: capacitación y entrenamiento. URL: <http://www.gestiopolis.com/canales/derrhh/articulos/no1/entrenamiento.htm> (consultado el 17 de septiembre de 2009)
- [26] Definicion, URL: <http://www.definicion.org> (consultado el 17 de septiembre de 2009)
- [27] Adobe Acrobat Connect Pro, URL: <http://www.adobe.com/products/acrobatconnectpro/> (consultado el 20 de septiembre de 2009)
- [28] eLecta, URL: <http://www.e-lecta.com/demo.asp> (consultado el 25 de septiembre de 2009)
- [30] Gartner, URL: http://gartner.com/it/products/mq/mq_ms.jsp (consultado el 30 de septiembre de 2009)
- [31] WiZiQ, URL: http://www.wiziq.com/Virtual_Classroom.aspx (consultado el 3 de octubre de 2009)
- [33] *Advanced Distributed Learning*, URL: <http://www.adlnet.gov/Pages/Default.aspx> (consultado el 5 de octubre de 2009)
- [37] Moodle, URL: <http://moodle.org> (consultado el 10 de octubre de 2009)
- [38] Wikipedia, URL: <http://es.wikipedia.org/wiki/Moodle> (consultado el 10 de octubre de 2009)
- [39] Instituto Politécnico Nacional, URL: <http://www.ipn.mx> (consultado el 11 de octubre de 2009)
- [40] Sistema de Campus Virtual Politécnico, URL: <http://148.204.103.84/moodle> (consultado el 13 de octubre de 2009)

- [52] IBM, URL: <http://www-01.ibm.com/software/rational/uml> (consultado el 7 de febrero de 2010)
- [54] Adobe Flex 3 (2010), URL: <http://www.adobe.com/es/products/flex/> (consultado el 12 de febrero de 2010)
- [55] Struts 2 (2010), URL: <http://struts.apache.org/2.1.6/index.html> (consultado el 16 de febrero de 2010)
- [56] JADEX (2010), URL: <http://vsis-www.informatik.uni-hamburg.de/projects/jadex/download.php> (consultado el 18 de marzo de 2010)
- [57] Jeremy Allaire, Macromedia Flash MX-A next generation rich client. Marzo 2002, URL: <http://www.adobe.com/devnet/flash/whitepapers/richclient.pdf> (consultado el 19 de marzo de 2010)
- [58] Adobe BlazeDS (2010), URL: <http://opensource.adobe.com/wiki/display/blazeds/BlazeDS> (consultado el 20 de marzo de 2010)
- [59] UltraVNC (2010), URL: <http://www.uvnc.com> (consultado el 30 de marzo de 2010)
- [60] Hibernate (2010), URL: <http://www.hibernate.org> (consultado el 1 de abril de 2010)
- [61] MySQL (2010), URL: <http://www.mysql.com> (consultado el 30 de marzo de 2010)
- [64] ESWA EXPERT SYSTEMS, URL: http://www.elsevier.com/wps/find/journaldescription.cws_home/939/description#description (consultado el 12 de julio de 2010)

ANEXO A

MANUAL DE OPERACIÓN

El presente anexo muestra el manual de operación de SEVirtual con información general de cómo utilizar SEVirtual para cada uno de los usuarios.

CONTENIDO

1. PRERREQUISITOS	144
2. INSTALACIÓN	145
3. MÓDULO DE LOGIN	145
4. MÓDULO DEL ADMINISTRADOR	146
4.1 REGISTRAR, ACTUALIZAR Y ELIMINAR USUARIOS	146
5. MÓDULO DEL PROFESOR	148
5.1 MENÚ DEL PROFESOR	148
5.2 ELIMINAR CURSO	149
5.3 EDITAR CURSO	150
5.4 NUEVO CURSO	150
6. MÓDULO DEL ALUMNO	162
6.1 MENÚ DEL ALUMNO	163
6.2 ELIMINAR CURSOS INSCRITOS	163
6.3 INSCRIBIR CURSO	164
6.4 VISUALIZAR CURSO INSCRITO	164
6.5 EVALUAR CURSO	172

1. PRERREQUISITOS

Los requisitos de instalación de SEVirtual, son los siguientes:

- Sistema operativo: multiplataforma.
- Máquina virtual de Flash (Flash Player) 9.0 o superior.
- Máquina virtual de Java (JRE) 5.0 o superior.
- Servidor VNC, para utilizar UltraVNC se necesita Windows XP o superior.
- MySQL.
- Navegador *Web*.

2. INSTALACIÓN

Para instalar SEVirtual es necesario realizar lo siguiente:

- Copiar la carpeta "db_sev" a la carpeta "data" del archivo de instalación de MYSQL.
- Copiar la carpeta "blazeds", "jadex-0.96" y "jadex-webbridge" al directorio raíz.
- Iniciar consola y en el directorio: "blazeds/tomcat/bin", ejecutar: startup.bat
- Iniciar consola y en el directorio: "blazeds/sampledb", ejecutar: startdb.bat
- Dar de alta en la variable de entorno: "CLASSPATH" los siguientes JARS que se encuentran en la carpeta: "jadex-0.96": jadex_rt.jar, jadex_standalone.jar, jadex_tools.jar, jibx-run.jar, xpp3.jar, jhall.jar y GroupLayout.jar.
- Iniciar consola y en el directorio raíz, ejecutar: java jadex.adapter.standalone.Platform
- Agregar a la plataforma de JADEX los siguientes JARS: webbridge_agent.jar, sevvirtualplan.jar, hibernate.jar, que se encuentran en la carpeta: "jadex-webbridge".
- Dentro de la plataforma JADEX desplegar: webbridge_agent.jar y dar de alta al Cordinator.xml.

3. MÓDULO DE LOGIN

En este módulo se muestra el acceso al sistema de los tres actores que existen: el administrador, el alumno y el profesor. La **Figura A.1 (a)** muestra la interfaz, los datos que son proporcionados para entrar al sistema son el correo y la contraseña del usuario, el sistema valida los datos del usuario y si existe identifica el tipo de usuario y carga su menú correspondiente. La **Figura A.1 (b)** presenta la interfaz cuando el correo y/o contraseña del usuario son incorrectos.

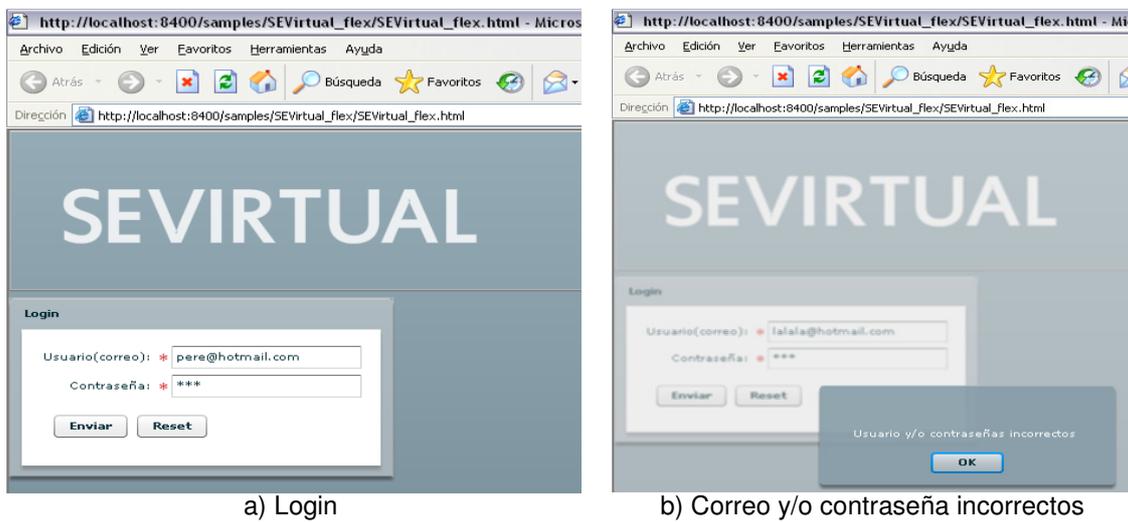


Figura A.1 Login

4 MÓDULO DEL ADMINISTRADOR

4.1 Registrar, actualizar y eliminar usuarios

Una vez que el administrador ha iniciado sesión dentro del sistema con su correo y contraseña correctos, se carga el menú del administrador. La **Figura A.2** muestra la interfaz del administrador, con las opciones de: registrar, actualizar y eliminar usuarios.

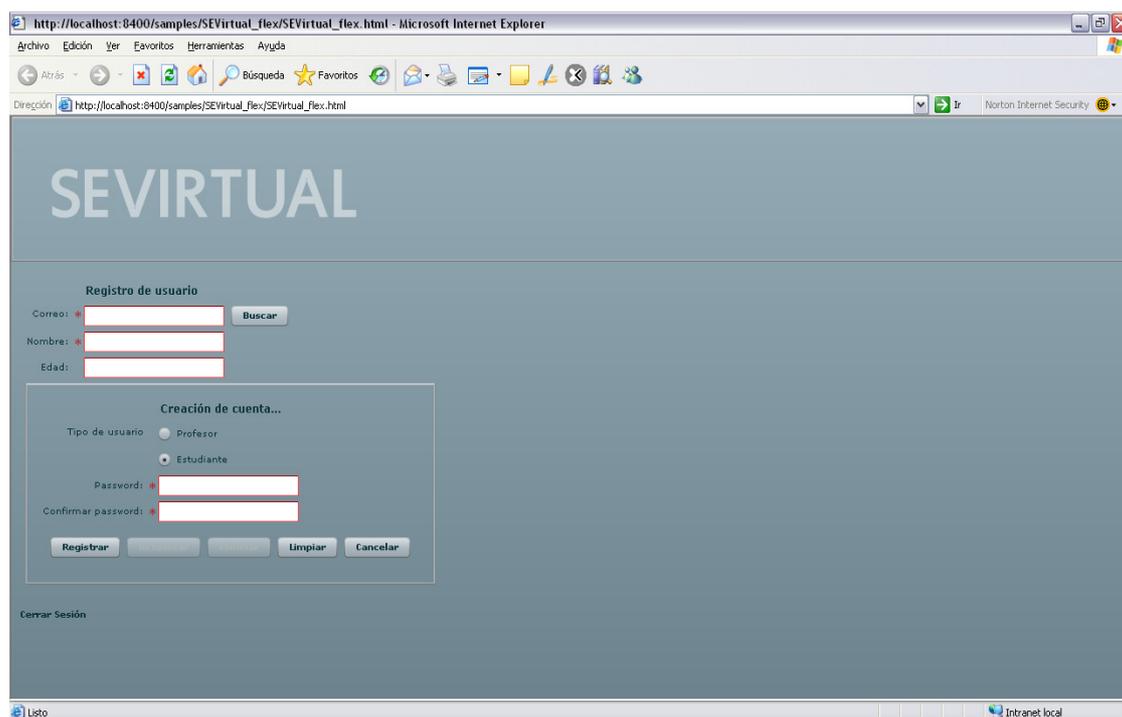
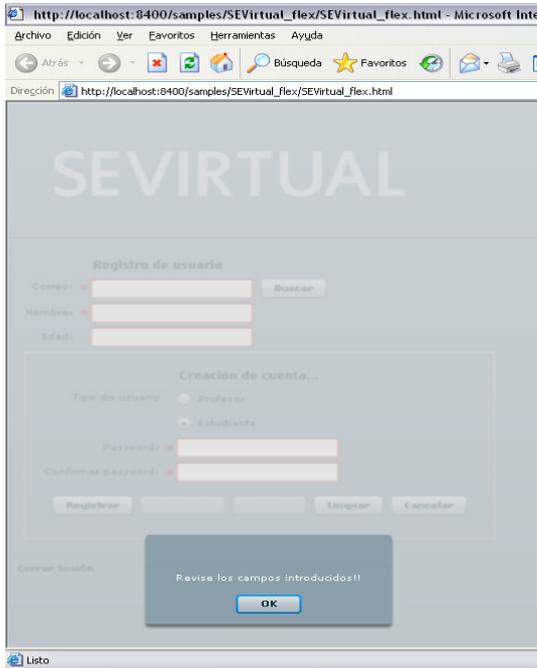


Figura A.2 Interfaz del administrador

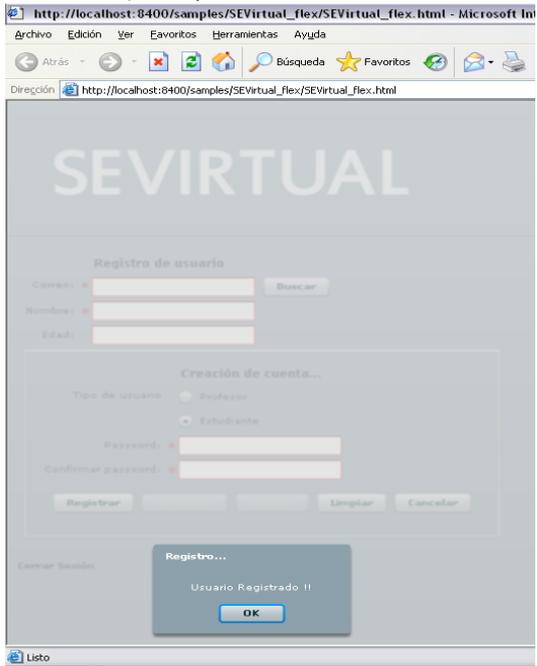
Para registrar a un usuario es necesario llenar todos los campos que son requeridos, es decir, los que tienen un asterisco, la **Figura A.3 (a)** muestra la interfaz que se despliega cuando no han sido introducidos los campos requeridos, mientras que en la **Figura A.3 (b)** se muestra la interfaz con los campos requeridos introducidos para su registro. La **Figura A.3 (c)** muestra el despliegue del mensaje de "Usuario Registrado" cuando el usuario se ha registrado con éxito. Para buscar algún usuario, el administrador deberá introducir el campo de correo y dar clic en el botón "Buscar", la **Figura A.3 (d)** muestra los datos desplegados de un usuario registrado anteriormente.



a) Campos no introducidos



b) Campos introducidos



c) Usuario registrado



d) Datos del usuario

Figura A.3. Registrar y buscar usuarios

Quando el administrador requiera modificar algún usuario, es necesario que busque al usuario con su correo y a continuación modifique los datos, la **Figura A.4 (a)** despliega la interfaz que se muestra cuando se han modificado los datos del usuario. Por último la **Figura A.4 (b)** presenta la interfaz cuando el usuario ha sido eliminado, previamente el usuario debió haber sido buscado.

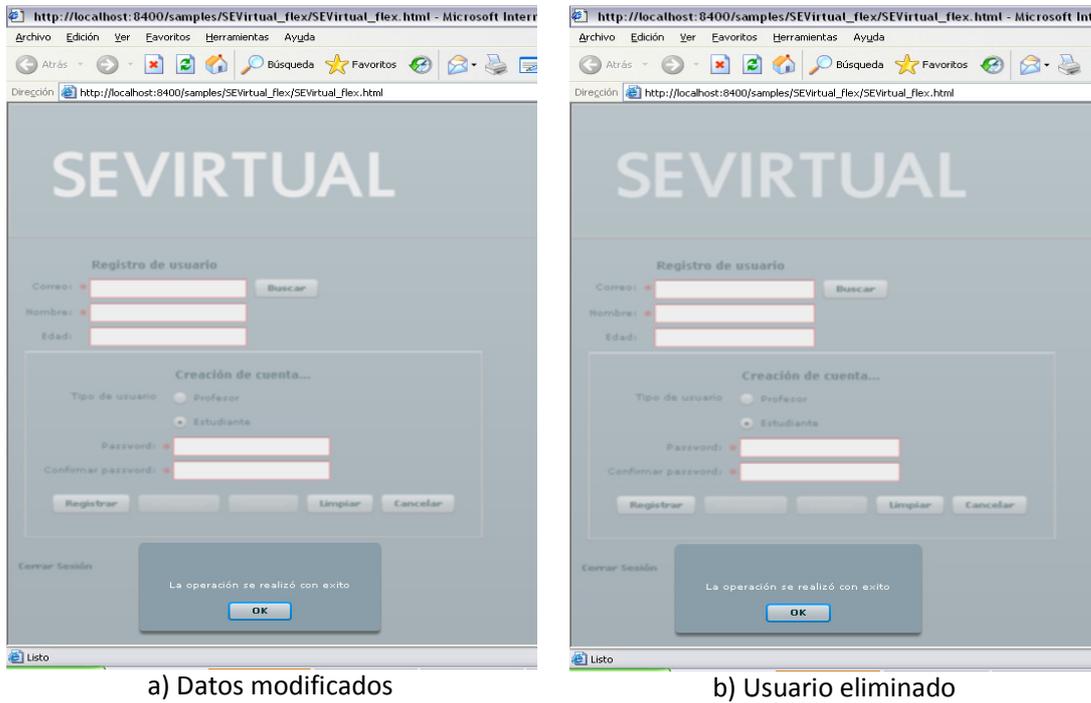


Figura A.4. Actualizar y eliminar usuarios

5. MÓDULO DEL PROFESOR

5.1 Menú del profesor

Una vez que el profesor ha iniciado sesión dentro del sistema con su correo y contraseña correctos, se carga el menú del profesor. La **Figura A.5** muestra el menú del profesor, con las opciones principales de: ver los cursos agregados, crear un nuevo curso y salir de la sesión.

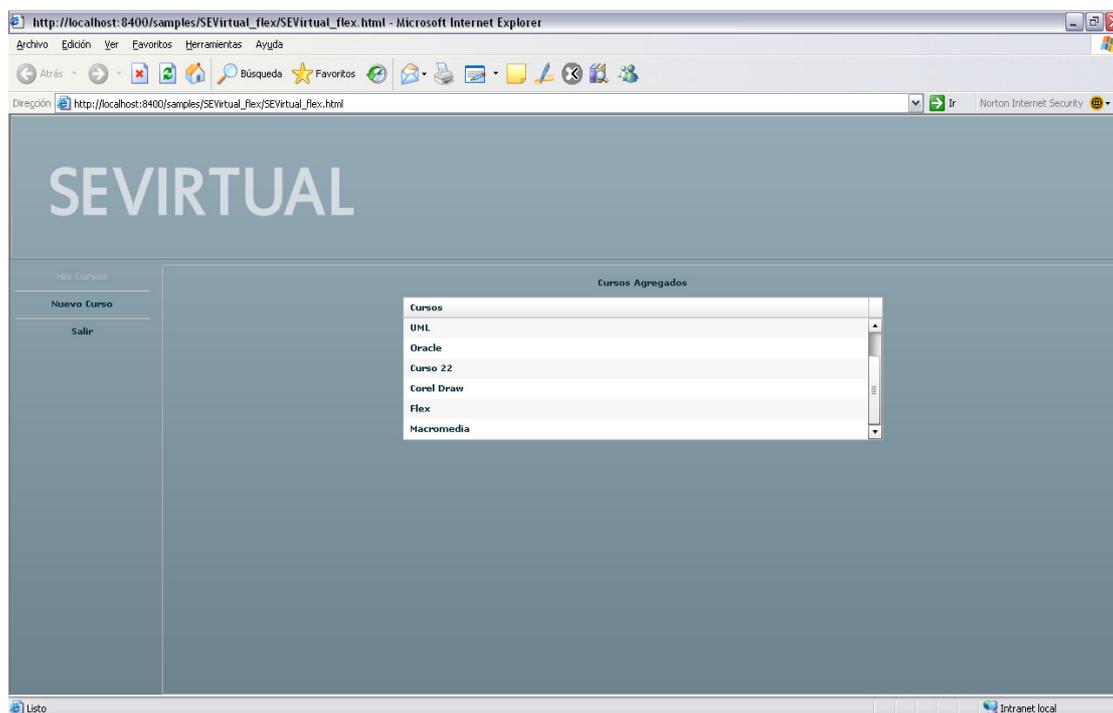


Figura A.5. Menú del profesor

5.2 Eliminar curso

La opción del menú que se carga al iniciar sesión es la de "Mis Cursos", donde se muestran todos los cursos que ha creado el profesor. Para eliminar algún curso se debe de seleccionar y presionar la tecla "Supr", la **Figura A.6** muestra el resultado de la eliminación del curso "Macromedia".

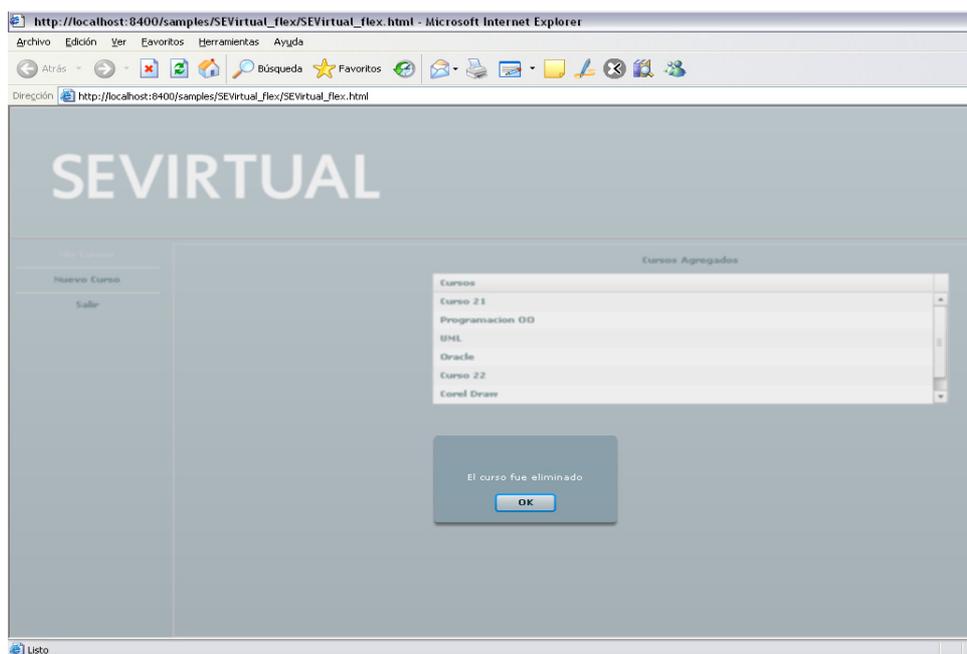


Figura A.6. Eliminar curso

5.3 Editar curso

Para la edición de cursos ya agregados, el profesor deberá dar doble clic sobre el curso, la **Figura A.7** presenta la interfaz de edición que se carga con los componentes que permiten editar el curso, esta interfaz es la misma que se muestra cuando se desea crear un nuevo curso, permitiendo la reusabilidad, por lo cual se mostrarán los resultados de estas pruebas en la parte de crear un nuevo curso que se explicará a continuación.

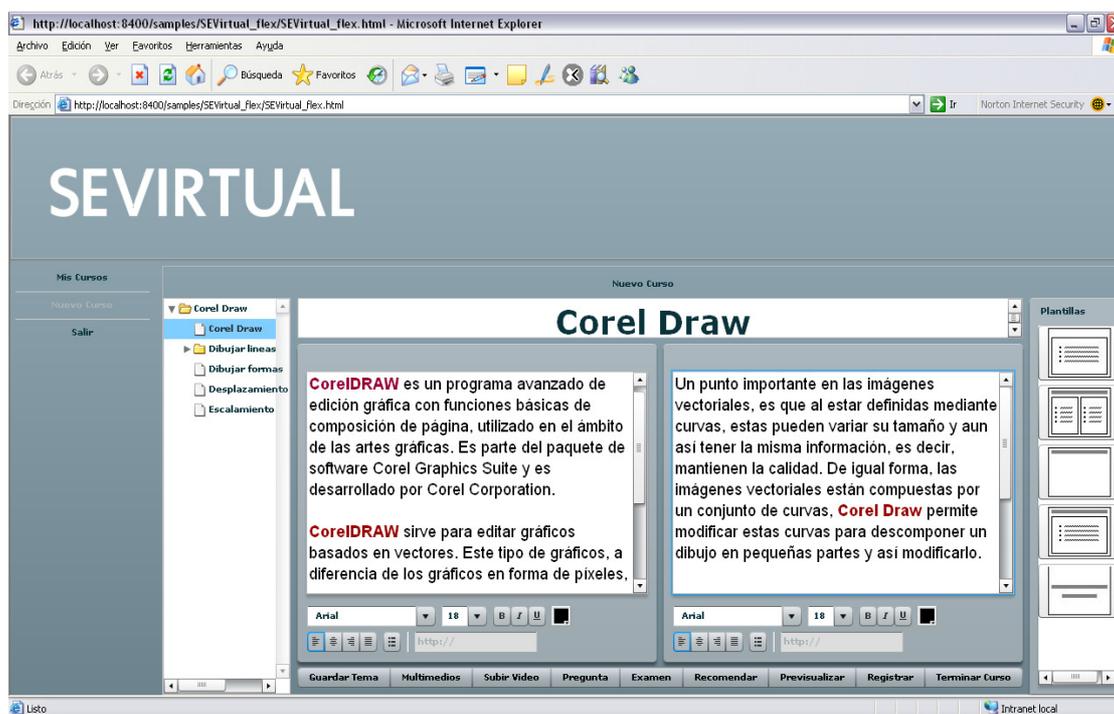
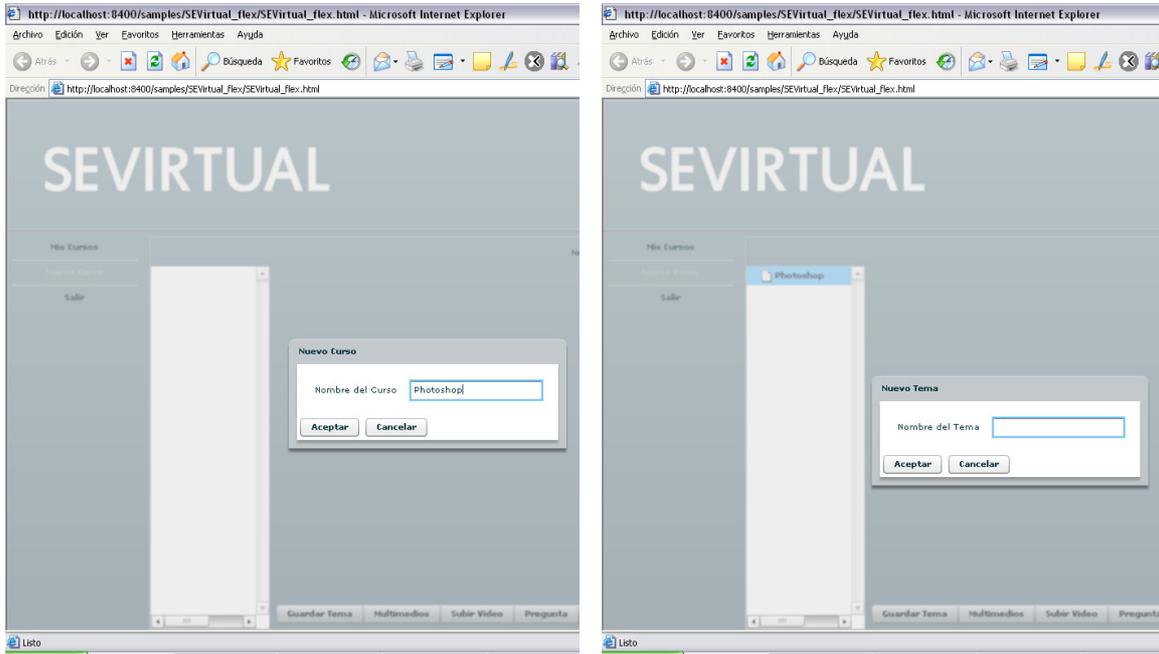


Figura A.7. Editar curso

5.4 Nuevo curso

En el menú del profesor, se debe seleccionar la opción "Nuevo Curso" para crear uno nuevo, la **Figura A.8 (a)** muestra la interfaz en la cual es necesario introducir el nombre del nuevo curso. Para agregar nuevos temas se debe seleccionar y dar "enter" en el elemento padre, la **Figura A.8 (b)** muestra el resultado de introducir un nuevo tema al elemento padre curso.



a) Introducir nombre del curso

b) Introducir nombre del tema

Figura A.8. Nuevo curso

Al agregar temas se va conformando la estructura del curso, la cual es una estructura de tipo abierta, para editar cada tema se puede seleccionar uno de los cinco componentes preconfigurados, como son: solo texto, titulo texto, titulo-subtitulo, titulo-doble texto o solo título. La **Figura A.9** muestra la edición del tema "Introducción" con el componente titulo doble texto, como se observa se puede seleccionar cualquier tipo de fuente, tamaño, estilo, etc.

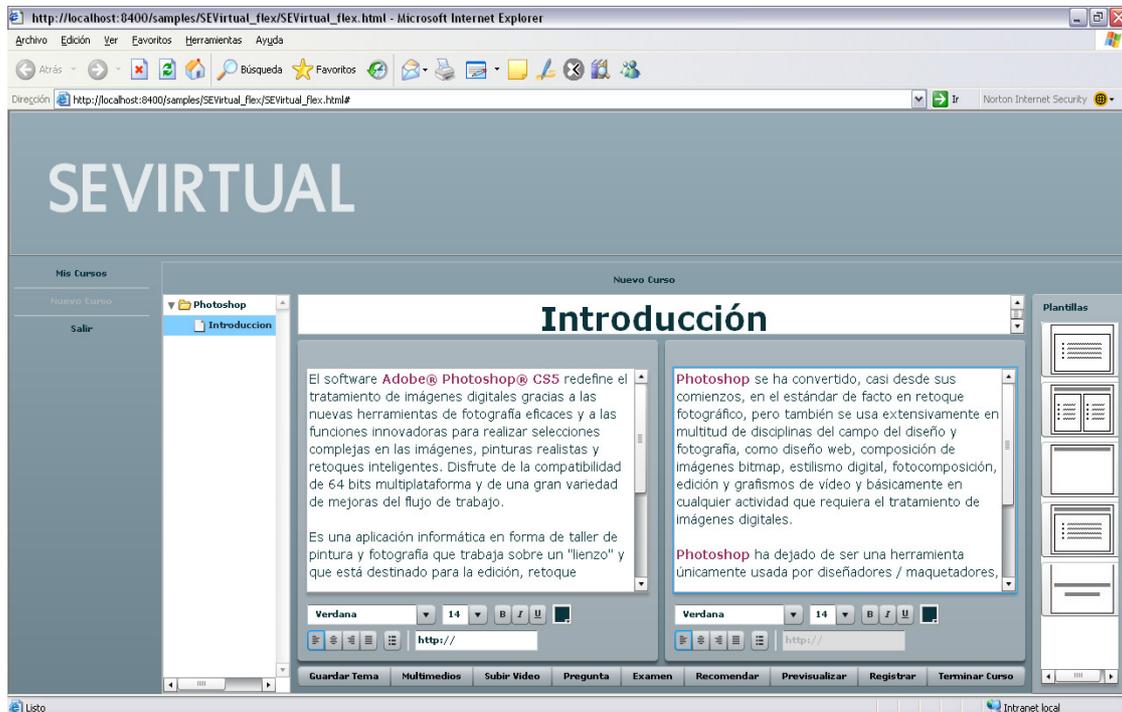


Figura A.9. Editar tema

Una vez editado, el tema debe ser guardado, para ello se oprime el botón **"Guardar Tema"**, esto con la finalidad de que los meta datos generado se envíen y guarde en el servidor, la **Figura A.10** muestra el resultado de guardar el tema.

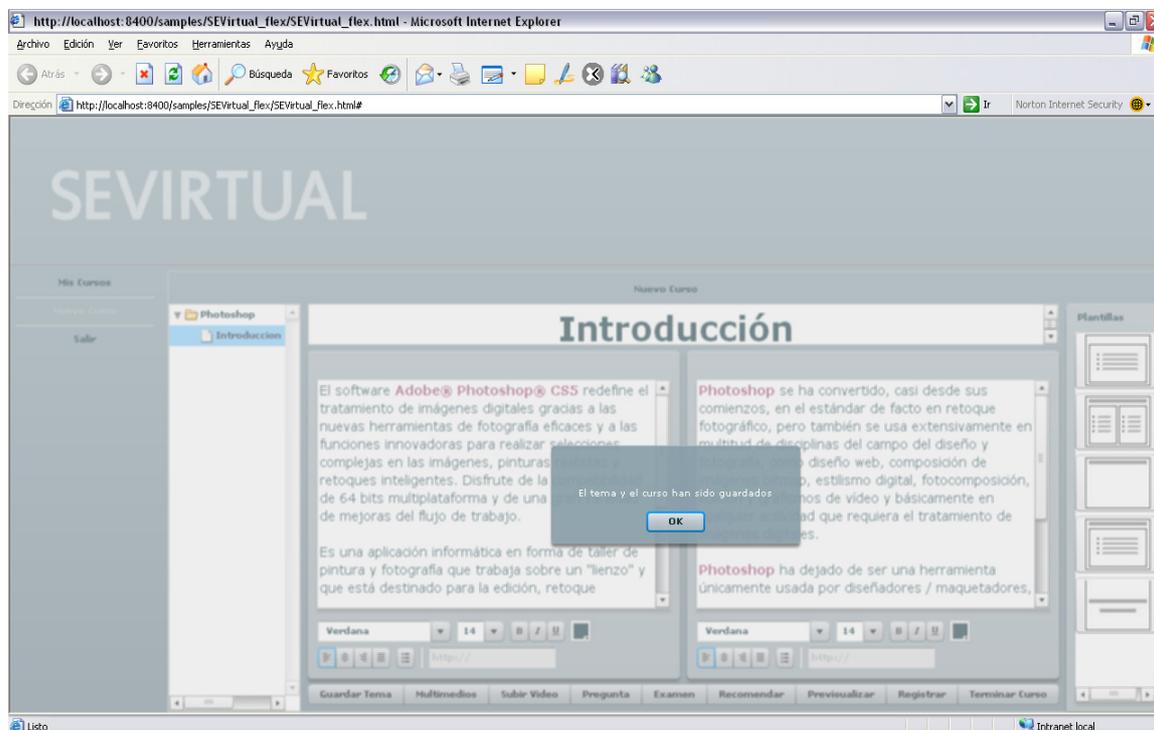
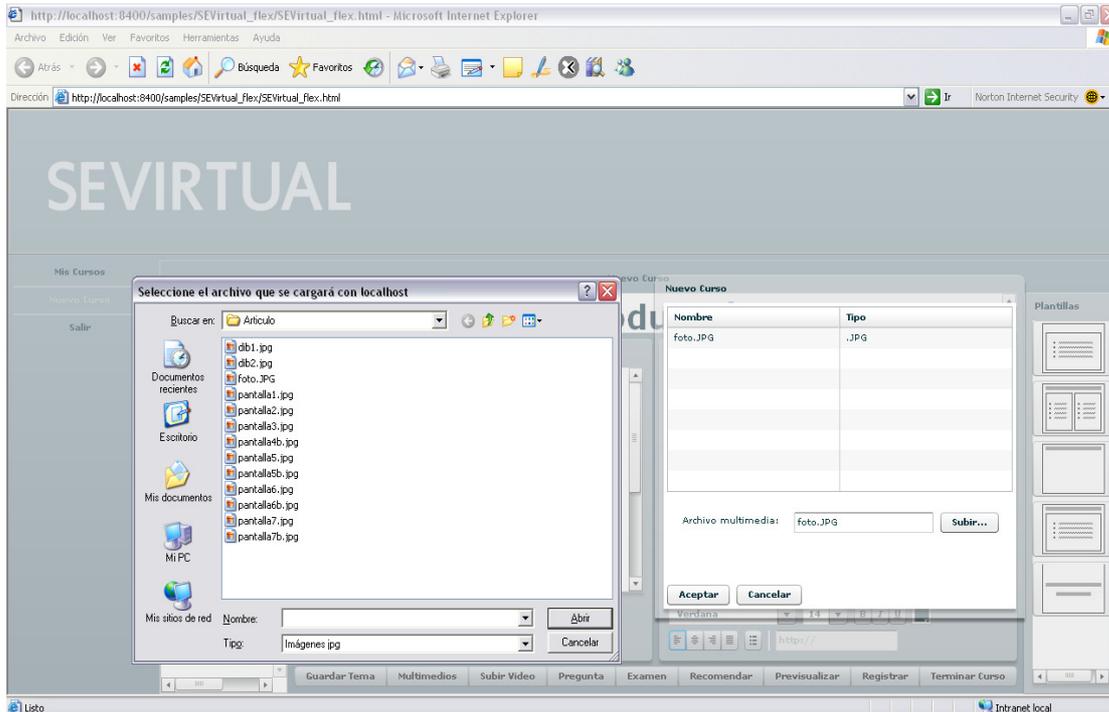
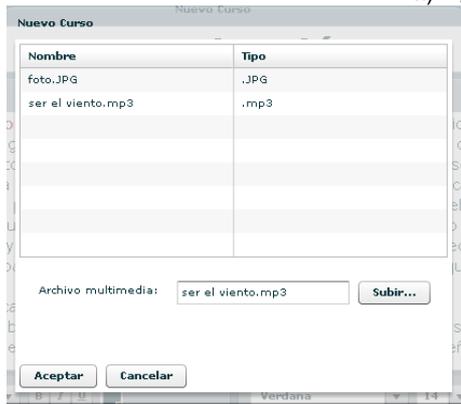


Figura A.10. Guardar Tema

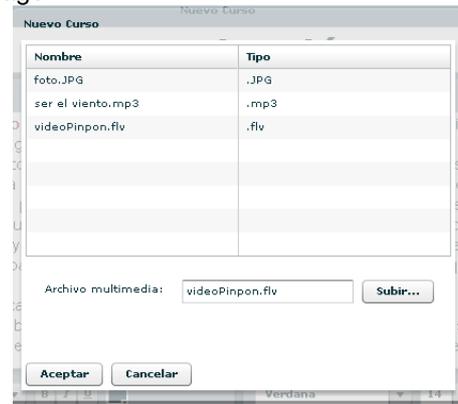
Cada tema puede tener uno o varios multimedios, para ello es necesario seleccionar el tema y dar clic en el botón **"Multimedia"**, la **Figura A.11 (a)** muestra el resultado de seleccionar una imagen JPG, en este proceso la imagen se guarda en el servidor. La **Figura A.11 (b)** muestra el resultado de subir un archivo de audio MP3 y la **Figura A.11 (c)** muestra el resultado de subir un archivo de video FLV.



a) Agregar imagen



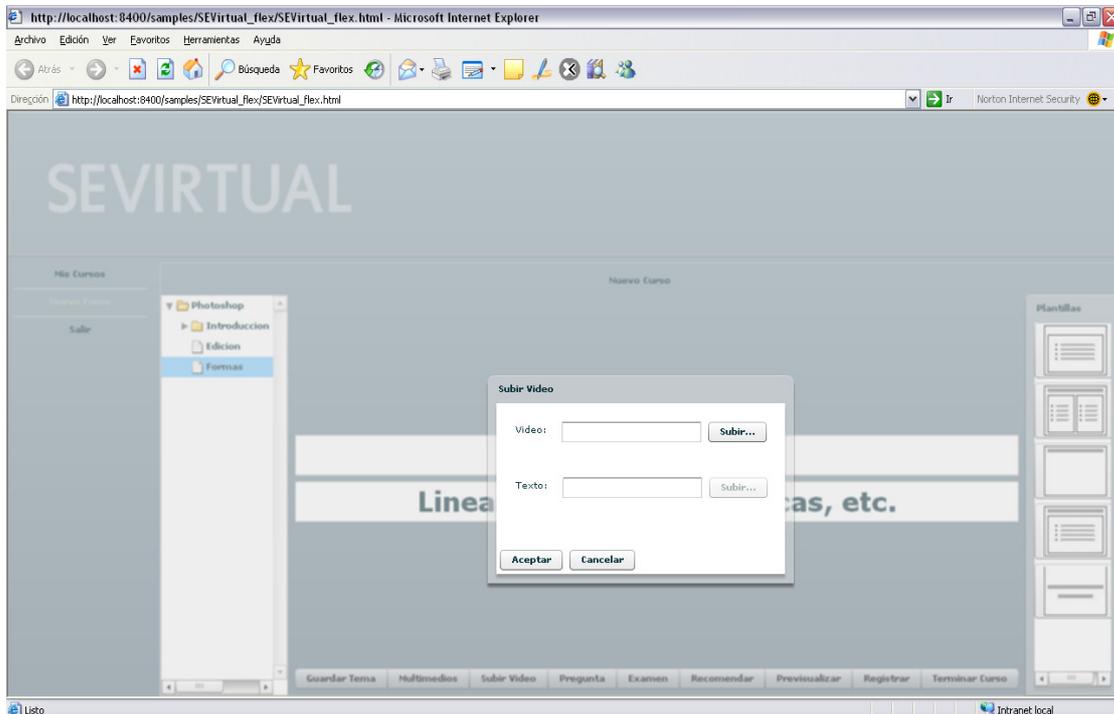
b) Agregar archivo MP3



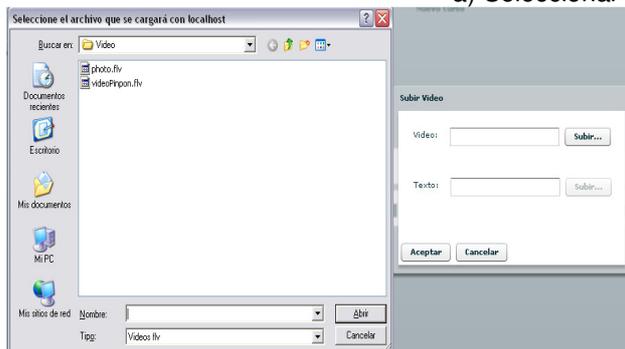
c) Agregar archivo FLV

Figura A.11. Agregar multimedia

El profesor puede subir un video explicativo del curso de entrenamiento, para ello es necesario dar clic en el botón "Subir Video", la **Figura A.12 (a)** muestra la interfaz que se despliega para subir el video. Al dar en el botón "Subir" el profesor puede seleccionar el video, la **Figura A.12 (b)** y **(c)** despliega el resultado de subir el video "photo.flv". También se puede subir un texto explicativo del video, para lo cual se da clic en el botón "Subir" del campo "Texto", la **Figura A.12 (d)** muestra el resultado de esta operación. Para sincronizar el contenido del curso con el video explicativo, una vez subido el video se despliega la interfaz que permite la sincronización, la **Figura A.12 (e)** muestra la interfaz en la cual el profesor selecciona "Cambiar Tema" durante la reproducción del video cada vez que sea necesario cambiar al siguiente tema, durante este procedimiento se generarán meta datos que contenga etiquetas con la información del tiempo en milisegundos en que se realiza cada cambio de tema.



a) Seleccionar video



b) Subir video "photo.flv"



c) Subir video "photo.flv"



d) Subir archivo de texto

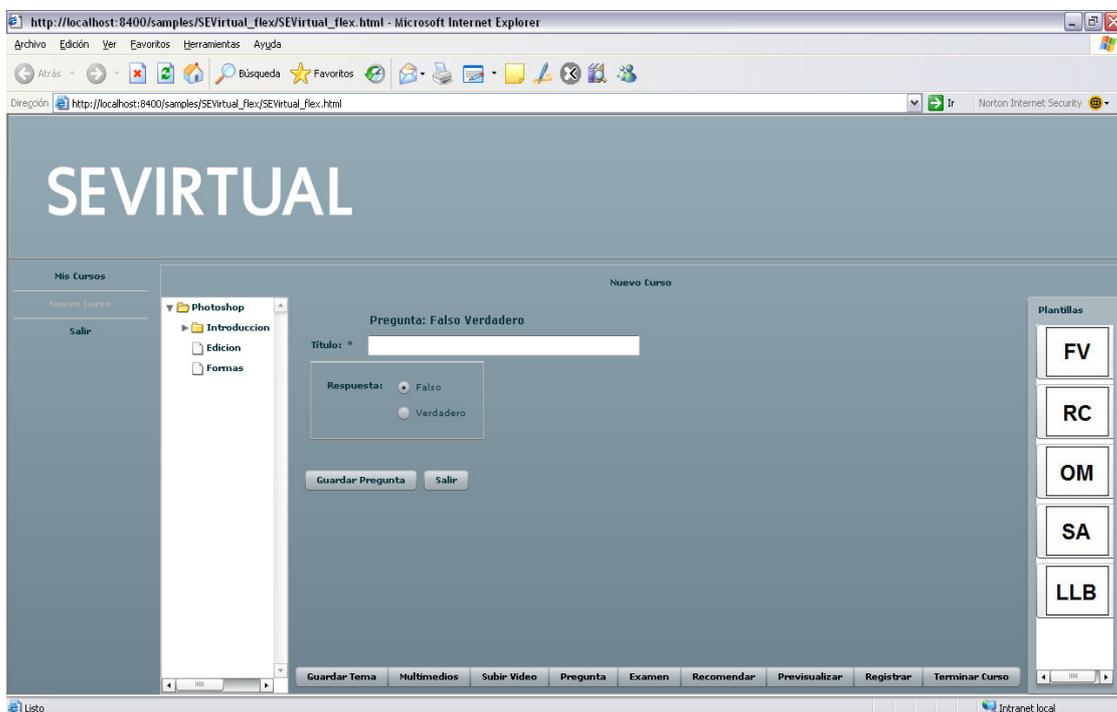


e) Sincronizar video y contenido

Figura A.12. Subir video explicativo del curso

El profesor puede crear preguntas de cinco tipos: falso/verdadero, relación de columnas, opción múltiple, llenado en blanco y selección de área, para lo cual debe dar clic en el botón "Pregunta", la **Figura A.13 a)** muestra la interfaz que se despliega. La pregunta que se carga al dar clic en el

botón "Pregunta" es la de falso/verdadero, las **Figuras A.13 (b)** y **(c)** muestran el resultado de agregar una pregunta de tipo falso/verdadero. La **Figura A.13 (d)** presenta el resultado que se despliega cuando no se selecciona un tema antes de guardar una pregunta. Las **Figuras A.13 (e)**, **(f)** y **(g)** se despliegan cuando se guarda una pregunta de tipo relación de columnas, en la primera interfaz es necesario dar clic en "Siguiente" para seleccionar el orden de las preguntas y opciones de respuesta como se observa en la **Figura A.13 (f)**. Las **Figuras A.13 (h)** e **(i)** muestran el resultado de guardar una pregunta de tipo opción múltiple, para esta pregunta se agregan las opciones de respuesta dando clic en el botón "Añadir" y se establece cual opción es la correcta a través de los *radio button*. Las **Figuras A.13 (j)** y **(k)** presentan la interfaz que se despliega cuando se agrega una pregunta del tipo selección de área. Por último las **Figuras A.13 (l)** y **(m)** muestran el resultado de guardar una pregunta del tipo llenado en blanco. Cada que se guarda una pregunta se genera un meta datos con la estructura de la pregunta correspondiente.



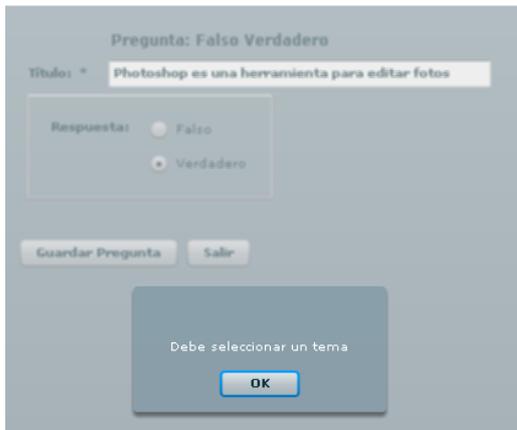
a) Agregar pregunta



b) Pregunta: falso verdadero



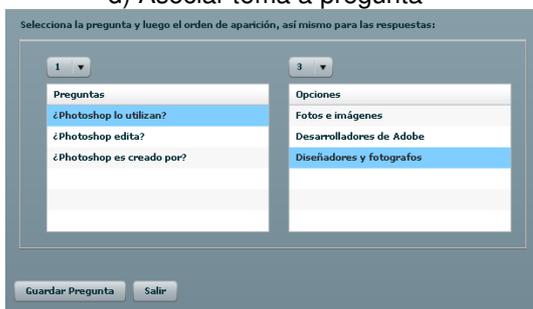
c) Guardar pregunta: falso verdadero



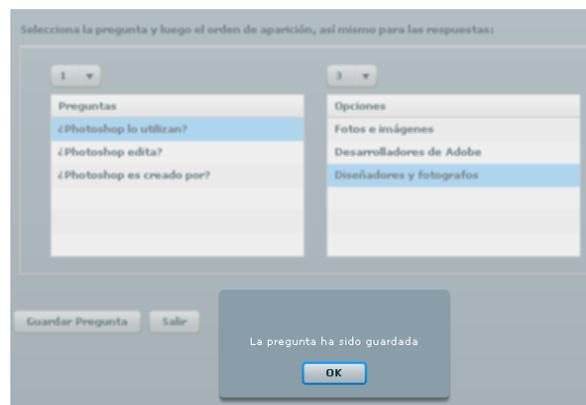
d) Asociar tema a pregunta



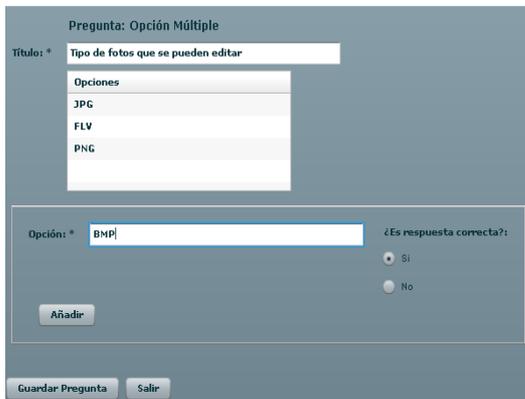
e) Pregunta: relación de columnas



f) Orden de las columnas



g) Guardar pregunta: relación de columnas



h) Pregunta: opción múltiple



i) Guardar pregunta: opción múltiple

j) Pregunta: selección de área

k) Guardar pregunta: selección de área

l) Pregunta: llenado en blanco

m) Guardar pregunta: llenado en blanco

Figura A.13. Agregar pregunta

El profesor puede crear un nuevo examen a partir de las preguntas que ha agregado, la **Figura A.14** despliega la interfaz que permite crear el examen. El examen puede ser previsualizado antes de guardarlo dando clic en el botón "Previsualizar", como se muestra en la **Figura A.15**. Para guardar el examen es necesario dar clic en el botón "**Guardar Examen**", con lo cual el meta datos generado con la estructura del examen se envía al servidor, la **Figura A.16** muestra el resultado de guardar el examen.

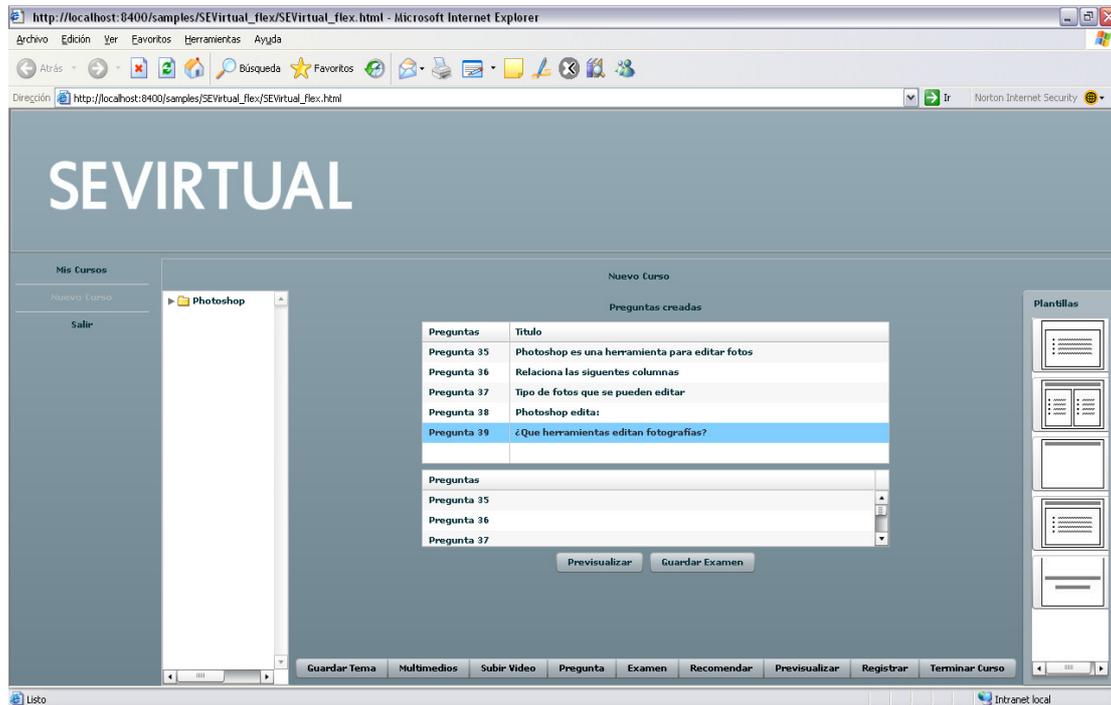


Figura A.14. Crear examen

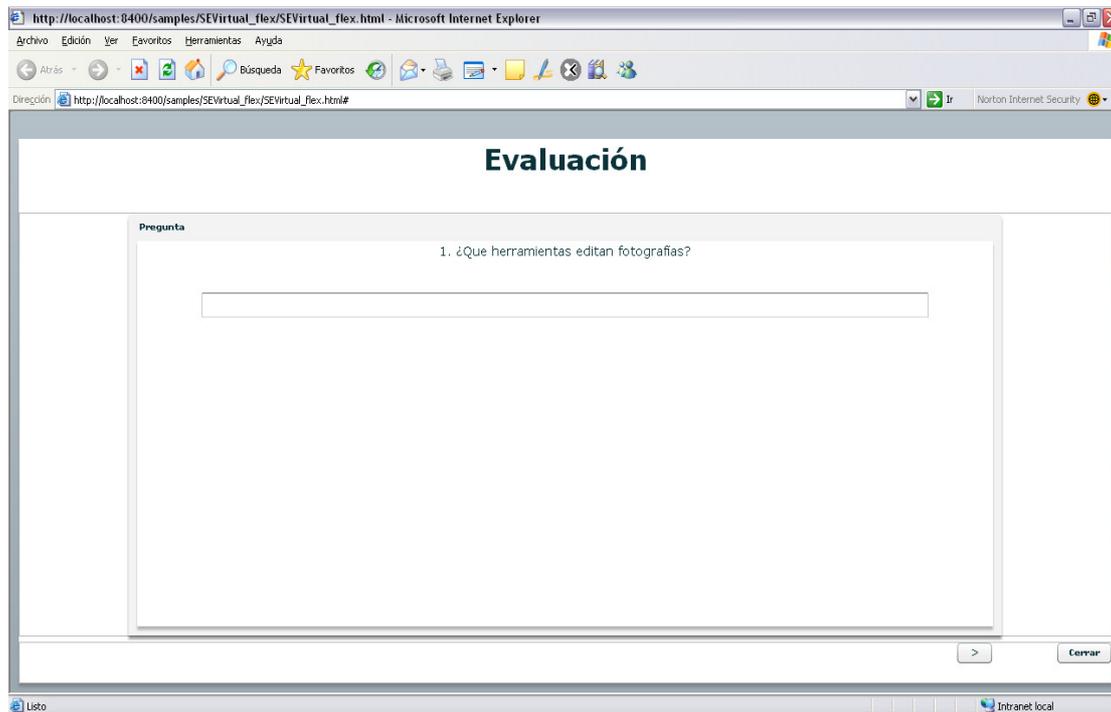


Figura A.15. Previsualizar examen

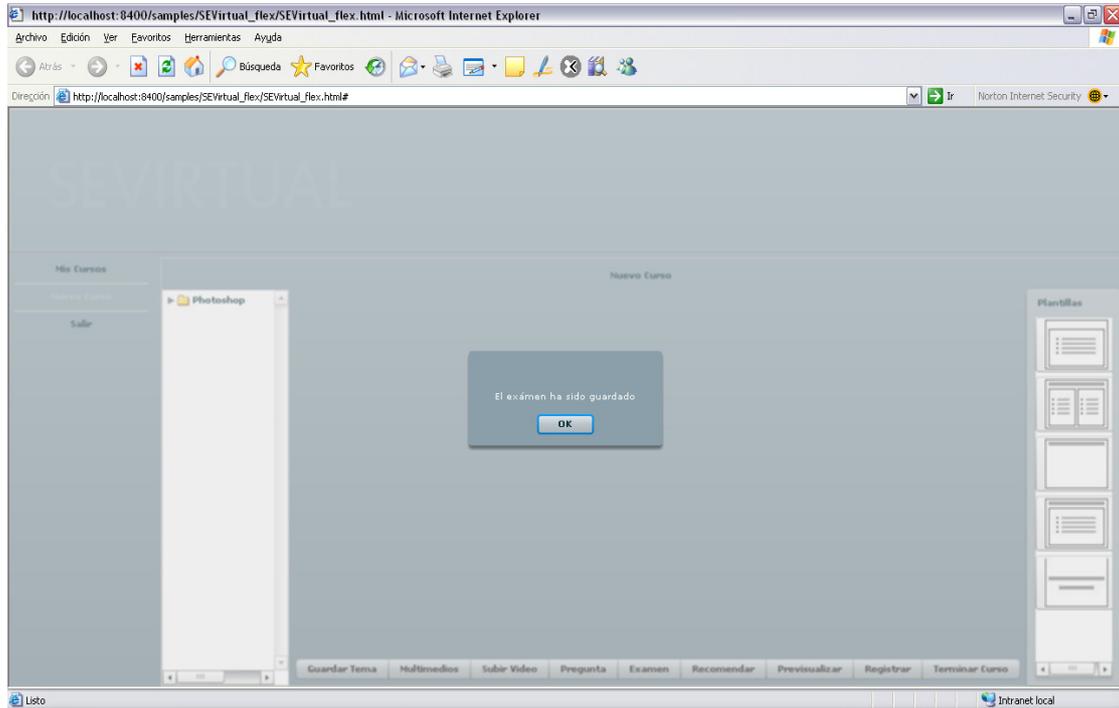
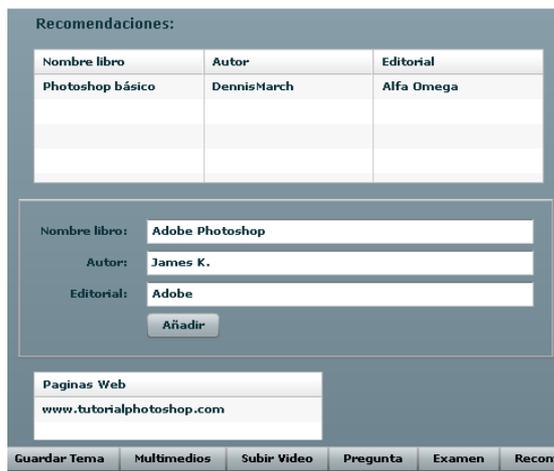


Figura A.16. Guardar examen

Una de las funciones del profesor virtual es la de dar recomendaciones al alumno, en base a los temas que se le han dificultado, identificados por medio de su evaluación, para ello es necesario que el profesor agregue recomendaciones de libros, páginas *Web* y ejercicios de cada tema, las recomendaciones se agregan seleccionando un tema y dando clic en el botón "Recomendar", la **Figura A.17 (a)** muestra la interfaz para realizar esta operación y la **Figura A.17 (b)** muestra el resultado de guardar las recomendaciones. La **Figura A.17 (c)** muestra el mensaje que se despliega al no seleccionar un tema y querer guardar una recomendación.



a) Recomendaciones



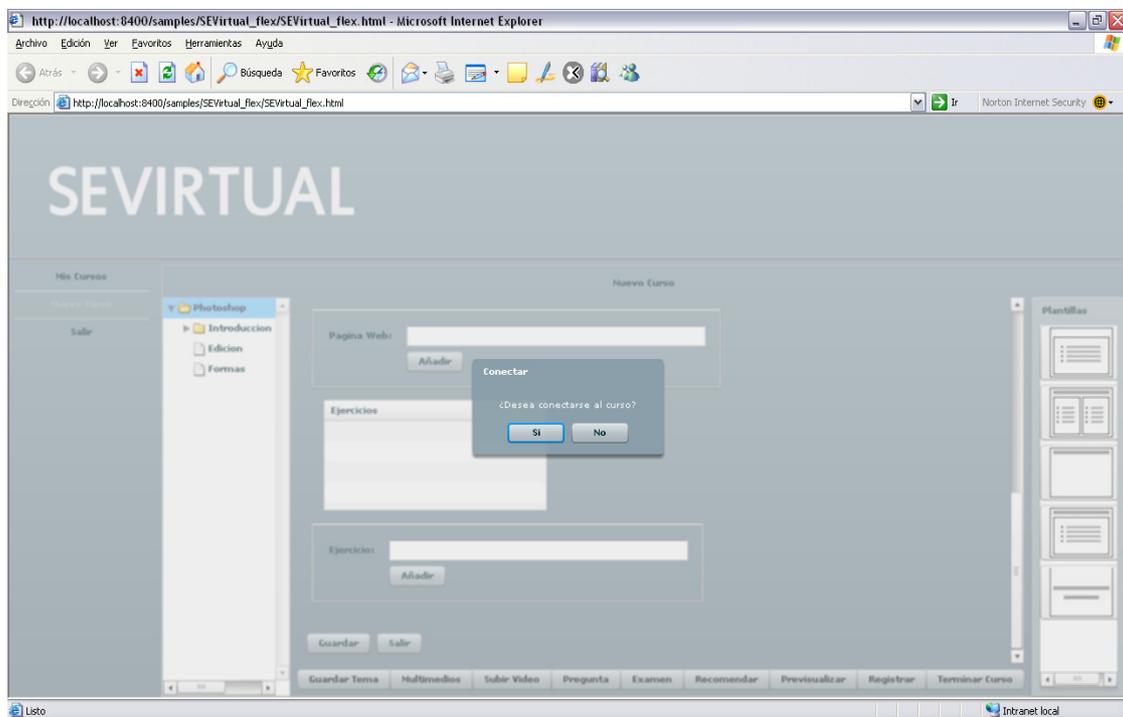
b) Recomendaciones guardadas



c) Asociar recomendación a tema

Figura A.17. Recomendar

El profesor puede previsualizar el curso de entrenamiento que está editando, el componente que se despliega es el mismo que ve el estudiante al visualizar algún curso al que se ha inscrito, posibilitando la reusabilidad de componentes. La **Figura A.18** muestra la interfaz que se despliega al dar clic en el botón "Previsualizar"; el sistema le pregunta al profesor si desea conectarse al curso, esto con la finalidad de darle la opción al profesor de interactuar en algún curso registrado.

**Figura A.18. Conectar al curso**

La **Figura A.19** muestra la interfaz que se despliega independientemente de si se encuentra conectado el profesor al curso, permitiendo su previsualización.

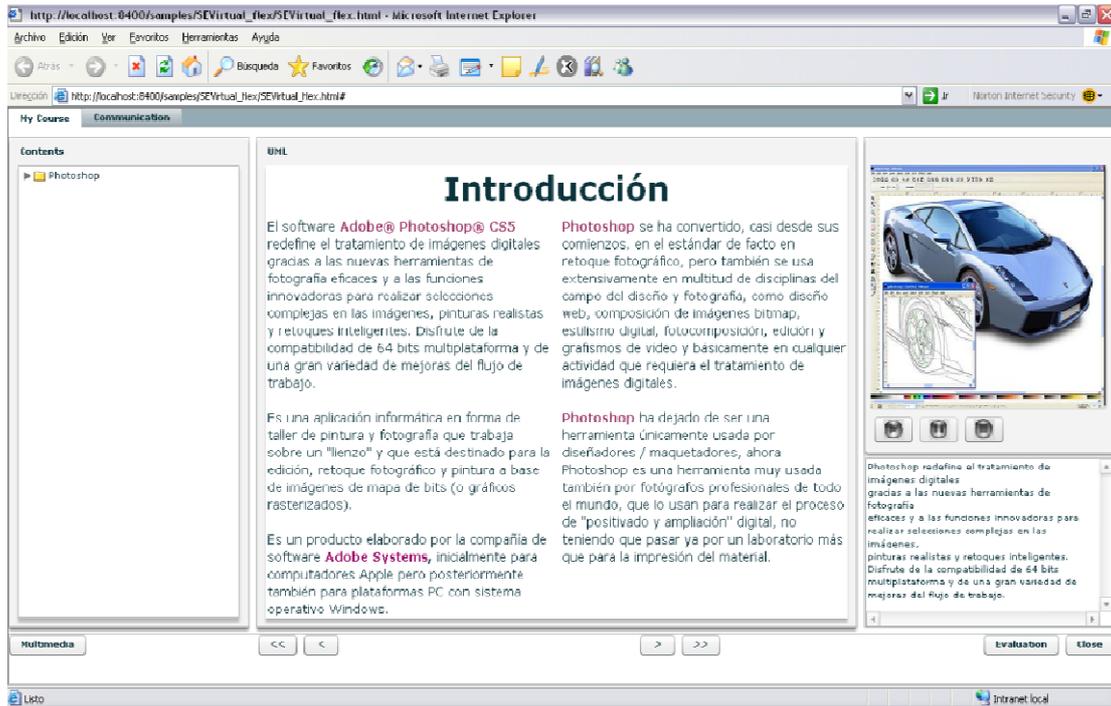


Figura A.19. Previsualizar curso

Para que el alumno pueda inscribirse a un curso, es necesario que el profesor lo registre, lo cual indica que el curso está listo para que el alumno lo pueda visualizar, para ello el profesor debe dar clic en el botón "Registrar". La **Figura A.20** muestra el resultado de registrar el curso.

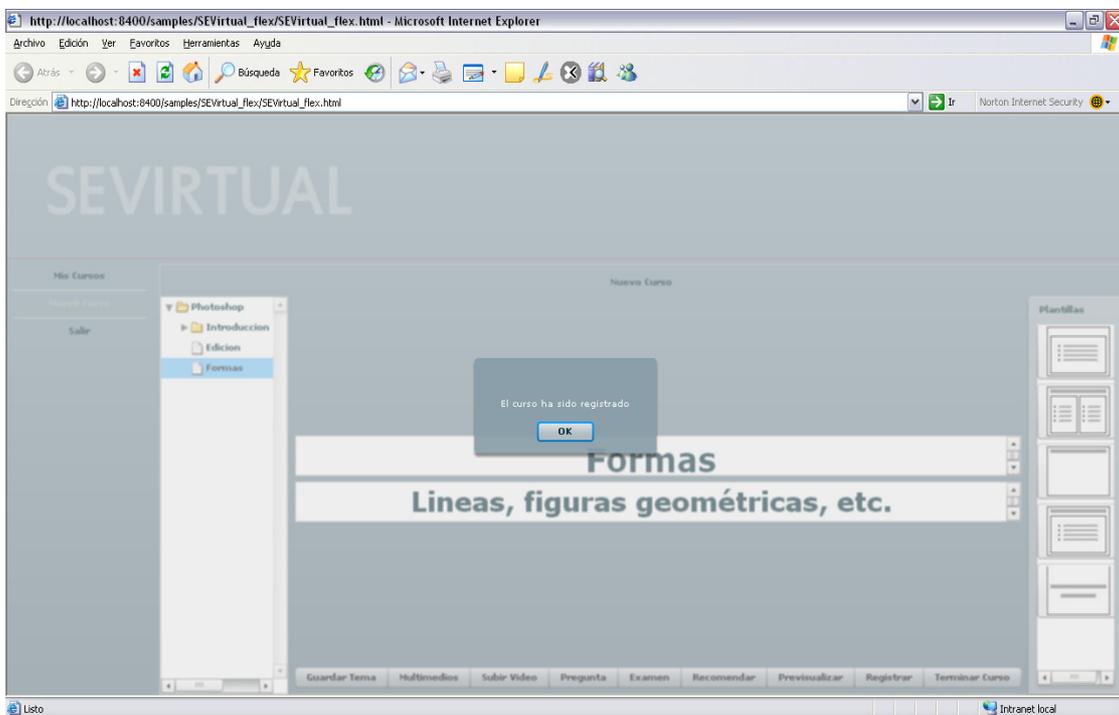


Figura A.20. Registrar el curso

Para terminar la edición de un curso, el profesor debe dar clic en el botón "Terminar Curso", para que le permita agregar más cursos y limpiar el área de edición. La **Figura A.21** muestra el resultado de terminar el curso.

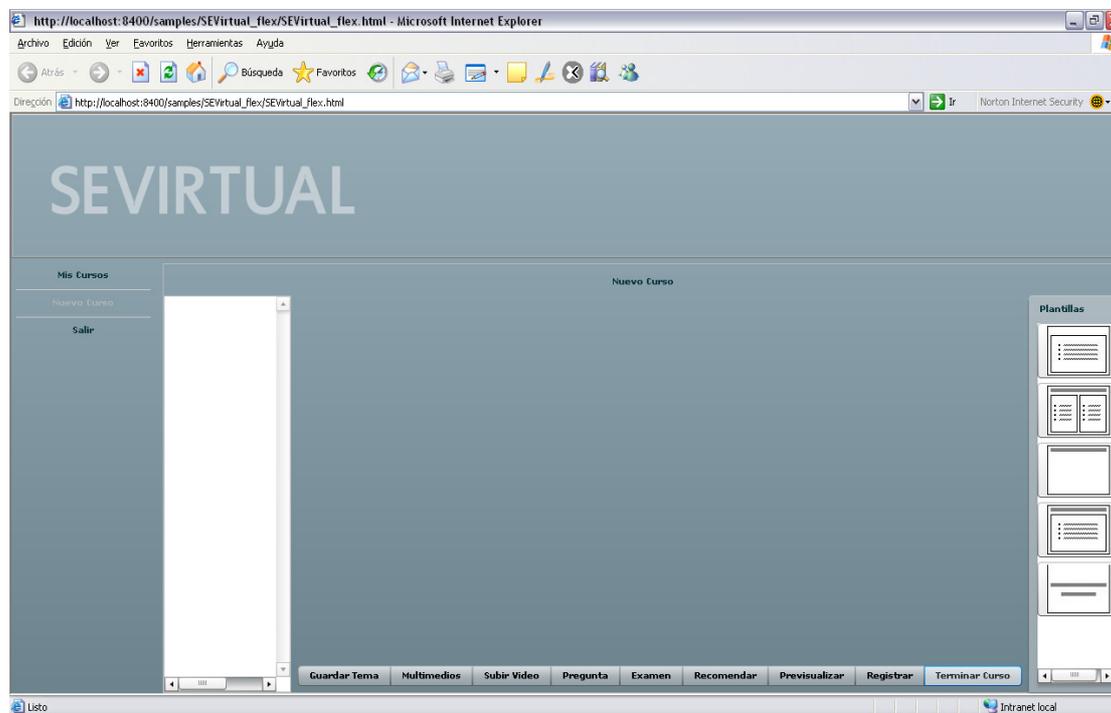


Figura A.21. Terminar el curso

6. MÓDULO DEL ALUMNO

Una vez que el alumno ha iniciado sesión dentro del sistema con su correo y contraseña correctos, se carga el menú del alumno. La **Figura A.22** muestra el menú del alumno, con las opciones principales: ver los cursos a los que se ha inscrito, inscribirse a algún curso, ver los cursos que el profesor virtual le ha reconfigurado y salir de la sesión.

6.1 Menú del alumno

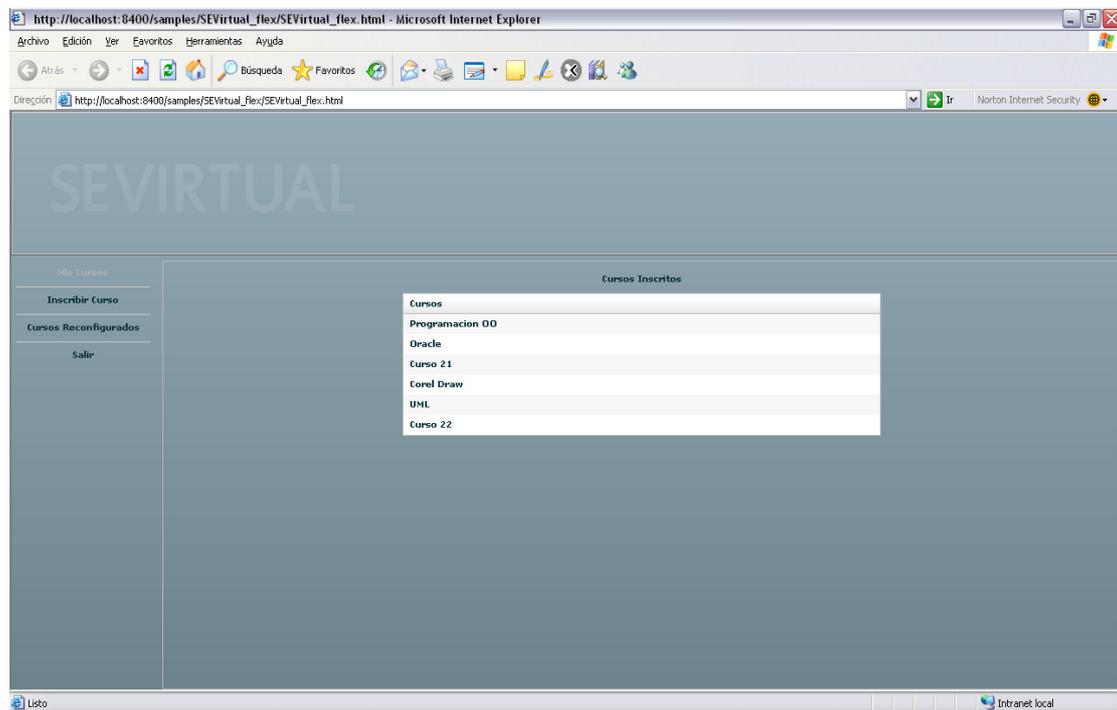


Figura A.22. Menú del alumno

6.2 Eliminar cursos inscritos

La opción del menú que se carga al iniciar sesión es la de "Mis Cursos", donde se muestran todos los cursos a los que se ha inscrito el alumno. Para eliminar algún curso se debe de seleccionar y presionar la tecla "Supr", la **Figura A.23** muestra el resultado de la eliminación del curso "Oracle".

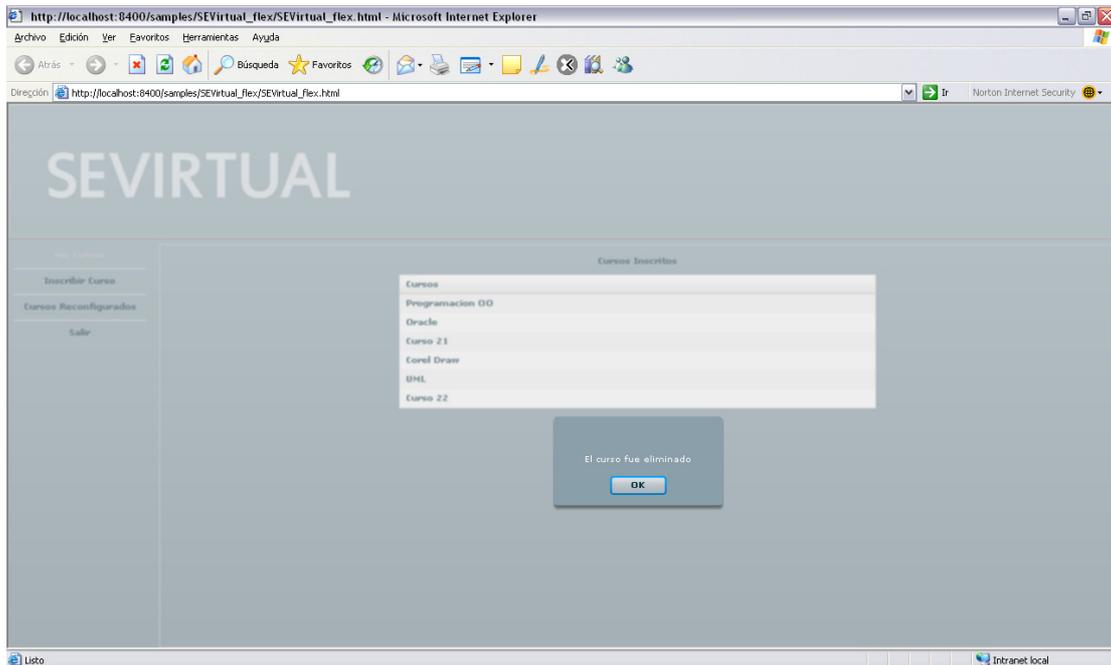


Figura A.23. Eliminar curso inscrito

6.3 Inscribir curso

En el menú del alumno, se debe seleccionar la opción “Inscribir curso” para visualizar la lista de cursos disponibles, para inscribirse a algún curso es necesario que el alumno presione doble clic sobre el curso al que se quiera inscribir. La **Figura A.24 (a)** muestra el resultado de inscribir el curso “Photoshop”. En la **Figura A.24 (b)** se despliega la lista actualizada de cursos inscritos al dar clic en la opción “Mis Cursos” del menú.



a) Inscribir curso: Photoshop

b) Lista de cursos inscritos

Figura A.24. Inscribir curso disponible

6.4 Visualizar curso inscrito

Para poder visualizar el curso, el alumno deberá dar doble clic sobre el curso que desee en la opción “Mis Cursos” del menú. La **Figura A.25** muestra el resultado que se despliega al visualizar el curso “Photoshop”.

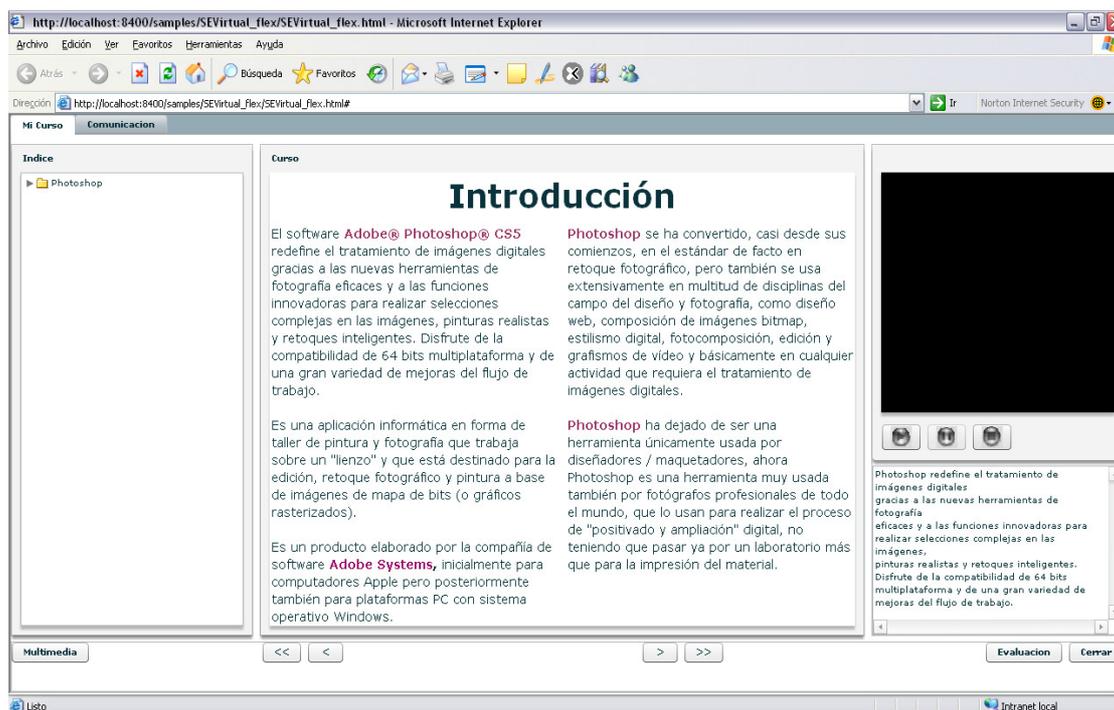


Figura A.25. Visualizar curso: Photoshop

Al cargar este componente se obtuvo la estructura del curso de los meta datos generados, cuando se cambia de tema, el componente de contenido descarga y carga el nuevo tema a través de la información de sus meta datos, permitiendo maximizar la reusabilidad. La estructura del curso se puede ver a través del componente índice que muestra el árbol con la estructura del curso, la **Figura A.26** despliega este componente.

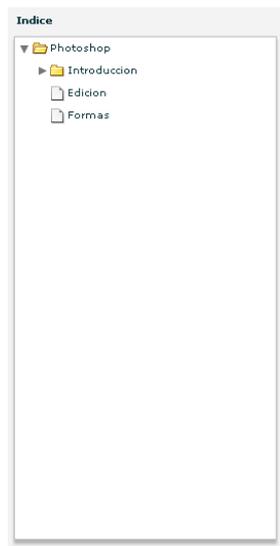


Figura A.26. Componente: índice

El usuario puede navegar a través del contenido del curso de varias formas, por medio del árbol de temas que se encuentra en el componente índice, por medio de la barra de navegación que se muestra en la **Figura A.27**, y por último a través del video explicativo del curso, con lo cual de

acuerdo a la sincronización que haya registrado el profesor en los temas se avanzará progresivamente.



Figura A.27. Componente: barra de navegación

La **Figura A.28** presenta el resultado de la exploración del curso a través del árbol de temas al seleccionar el tema "Formas".

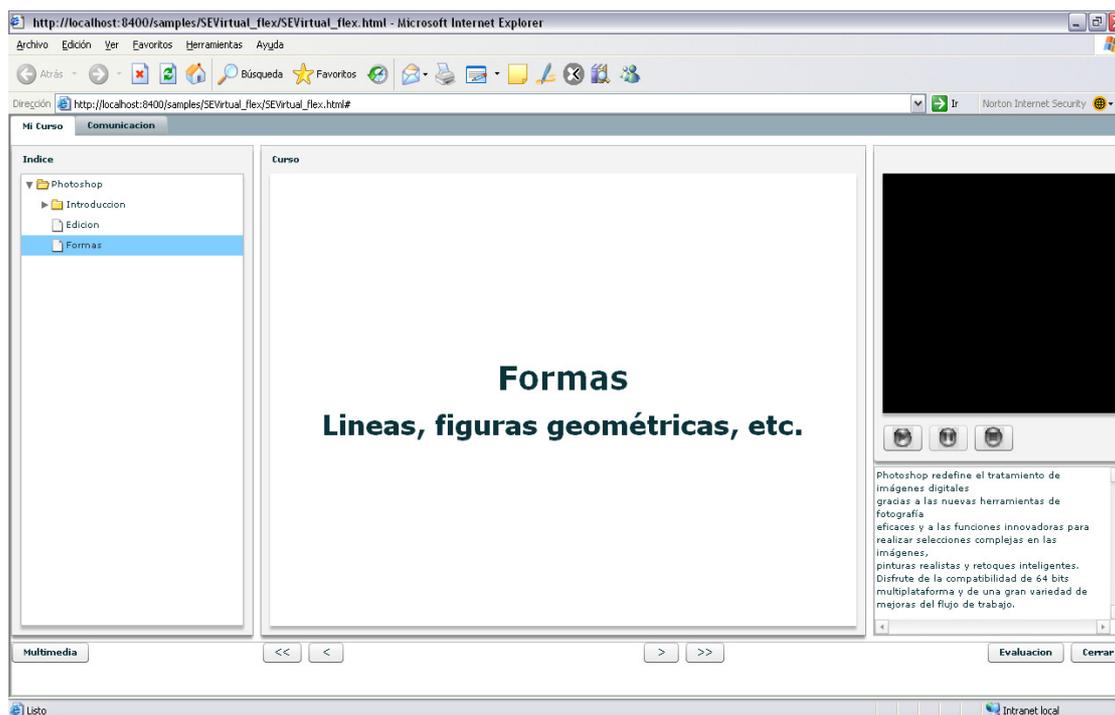


Figura A.28. Navegación a través del árbol de temas

La **Figura A.29** muestra el resultado de la navegación del curso a través de la barra de navegación, al dar clic en el botón anterior "<" se carga el tema "Edición".

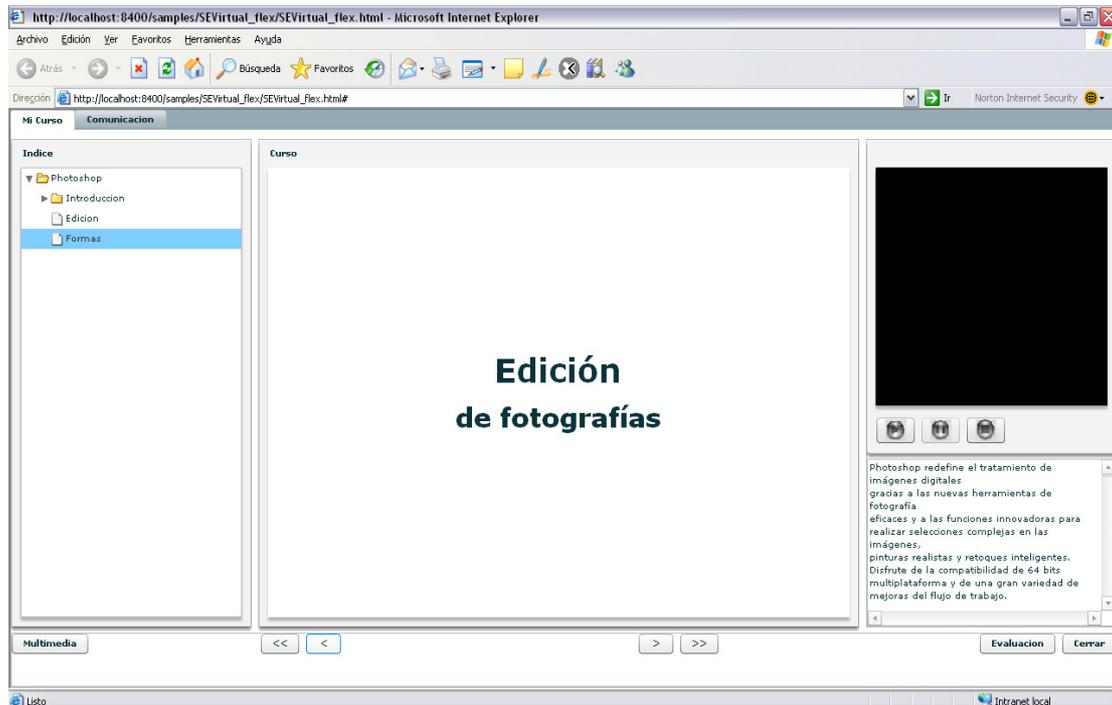


Figura A.29. Navegación a través de la barra de navegación

La **Figura A.30** despliega el resultado de la navegación automática del curso a través de la reproducción del video explicativo, donde los temas van avanzando automáticamente.

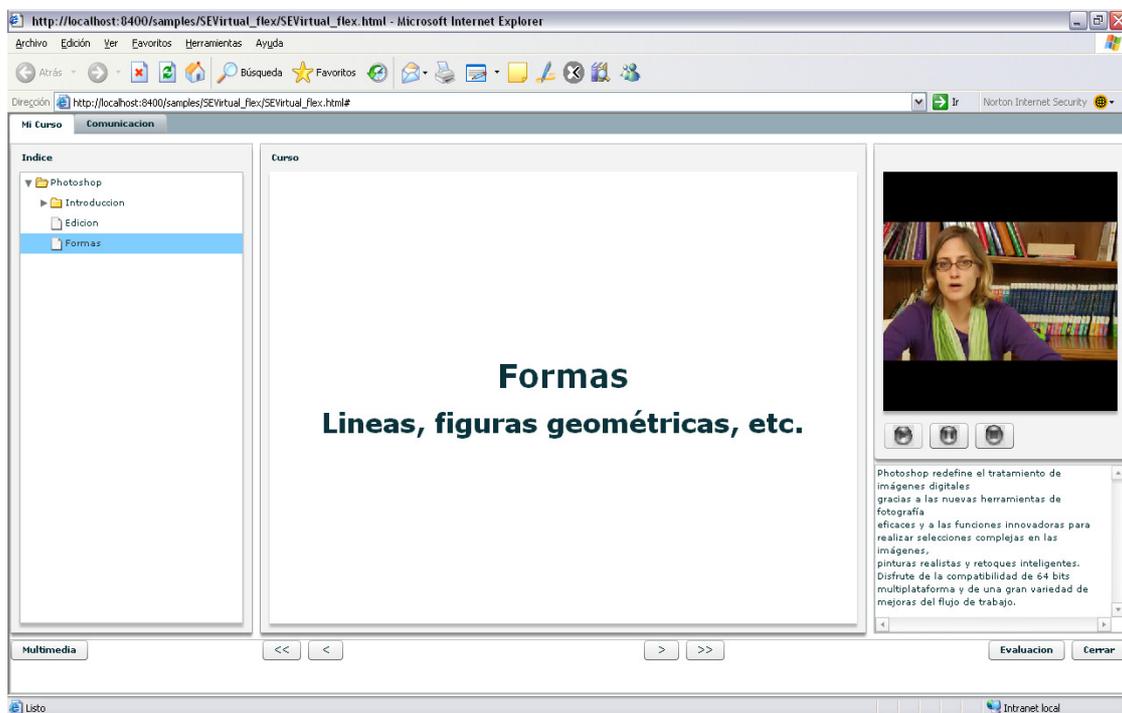
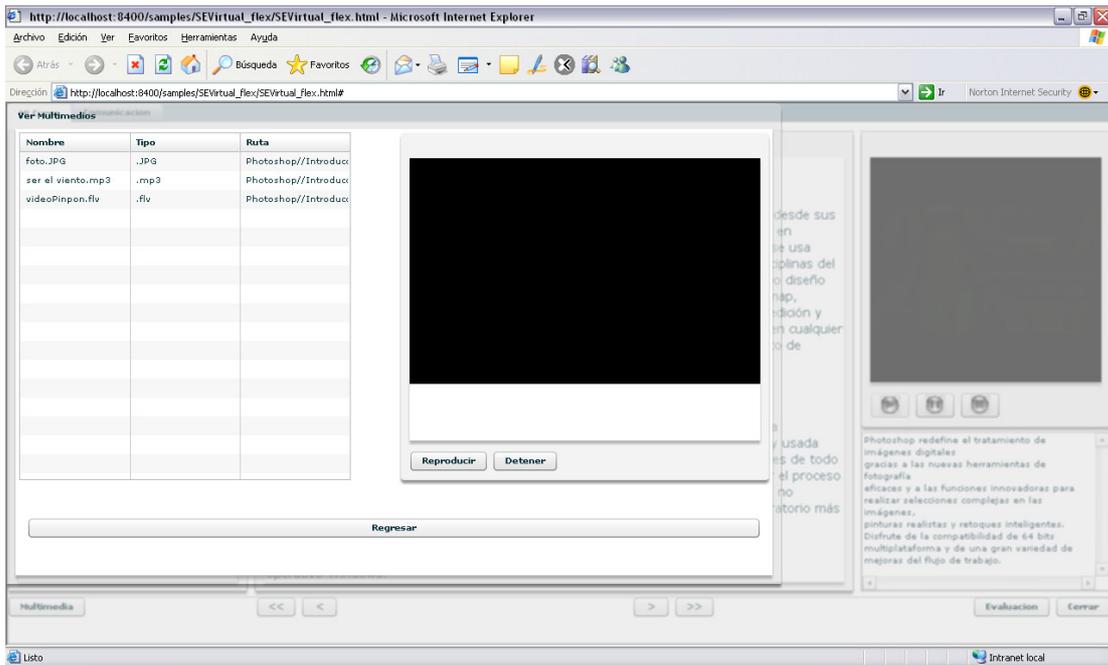
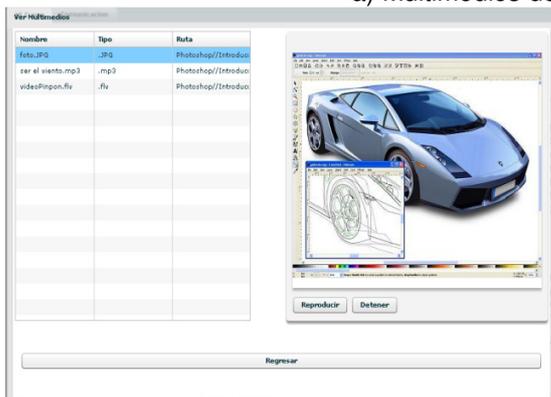


Figura A.30. Navegación a través de la reproducción del video

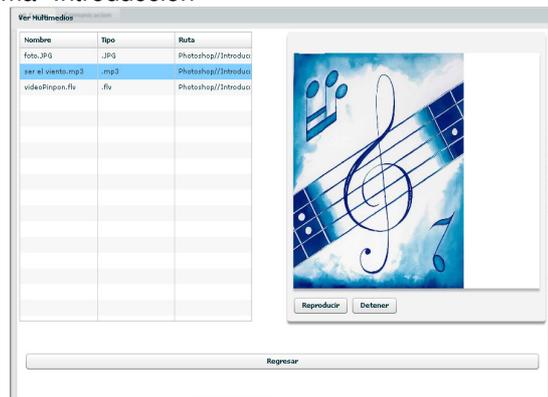
Para mostrar los multimedia agregados a algún tema, es necesario seleccionar el tema y dar clic en el botón "Multimedia", la **Figura A.31 (a)** despliega el resultado de seleccionar el tema "Introducción" y dar clic en el botón "Multimedia" para mostrar los multimedia que se han agregado a ese tema. Para el despliegue y reproducción de algún multimedia es necesario seleccionarlos. La **Figura A.31 (b)** presenta el resultado de seleccionar el multimedia "foto.JPG". La **Figura A.31 (c)** despliega el resultado de seleccionar el multimedia "ser el viento.mp3" con lo cual el audio empezará a reproducirse. La **Figura A.31 (d)** presenta el resultado de seleccionar el multimedia "videoPinpon.flv" comenzando su reproducción.



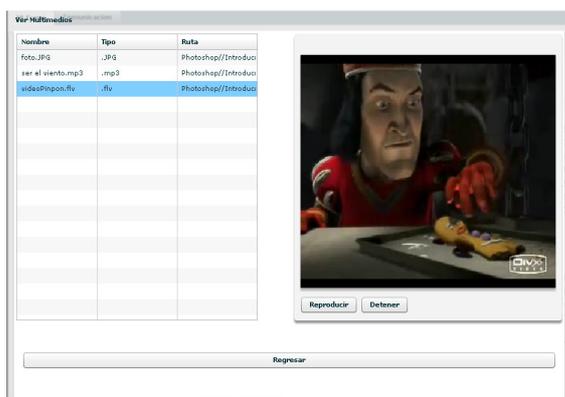
a) Multimedia del tema "Introducción"



b) Multimedia imagen



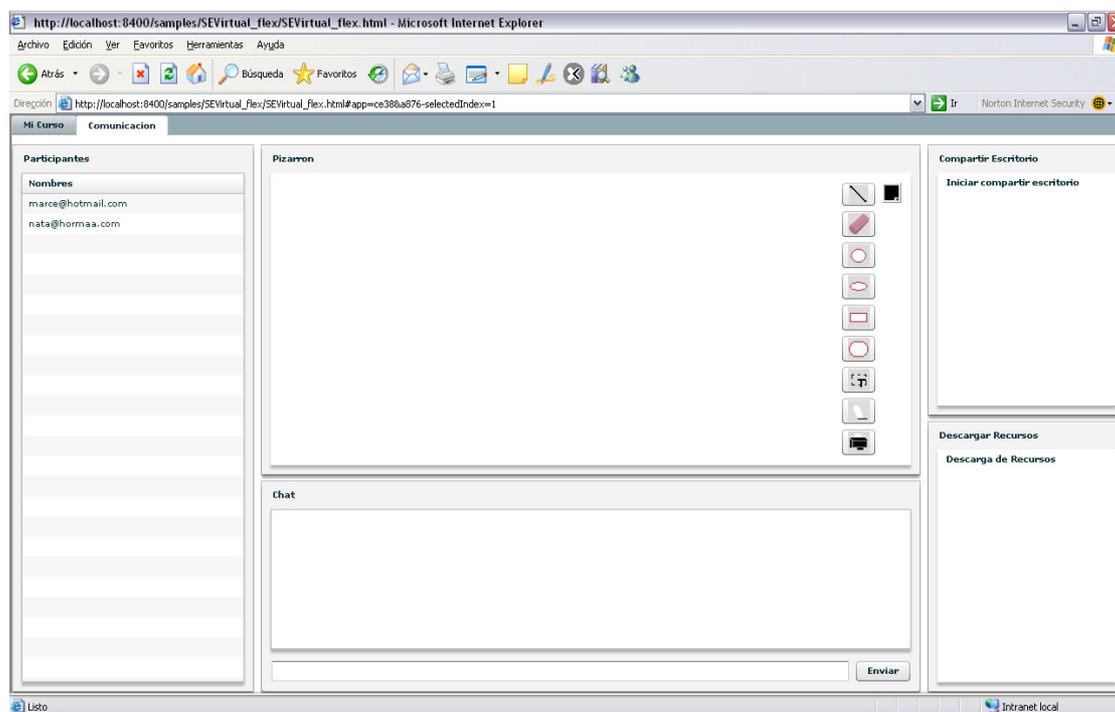
c) Multimedia audio



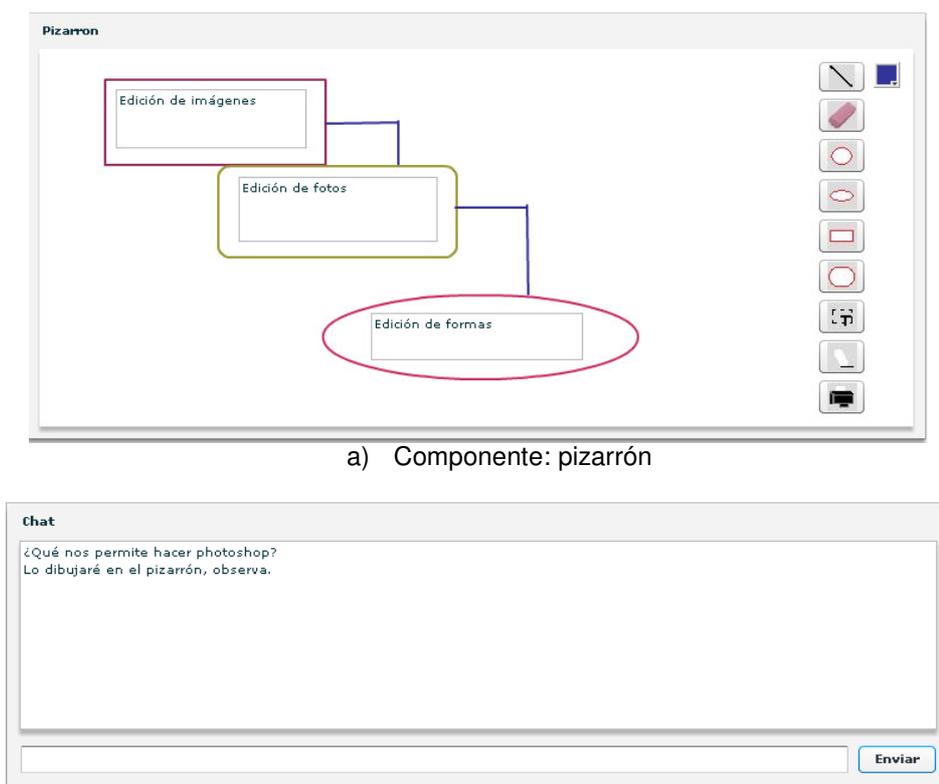
d) Multimedia video

Figura A.31. Componente: multimedia

Al dar clic en la pestaña "Comunicación" se cargan las herramientas colaborativas que permiten a los participantes colaborar y comunicarse, además se carga la lista de participantes del curso que están conectados en ese momento. La **Figura A.32** muestra esta interfaz.

**Figura A.32. Componente: comunicación**

La **Figura A.33 (a)** muestra el resultado de utilizar el pizarrón, a través del cual se comparte todo lo que se dibuje a todos los participantes que estén conectados. La **Figura A.33 (b)** muestra el resultado de enviar un mensaje de texto a los participantes a través del componente chat.

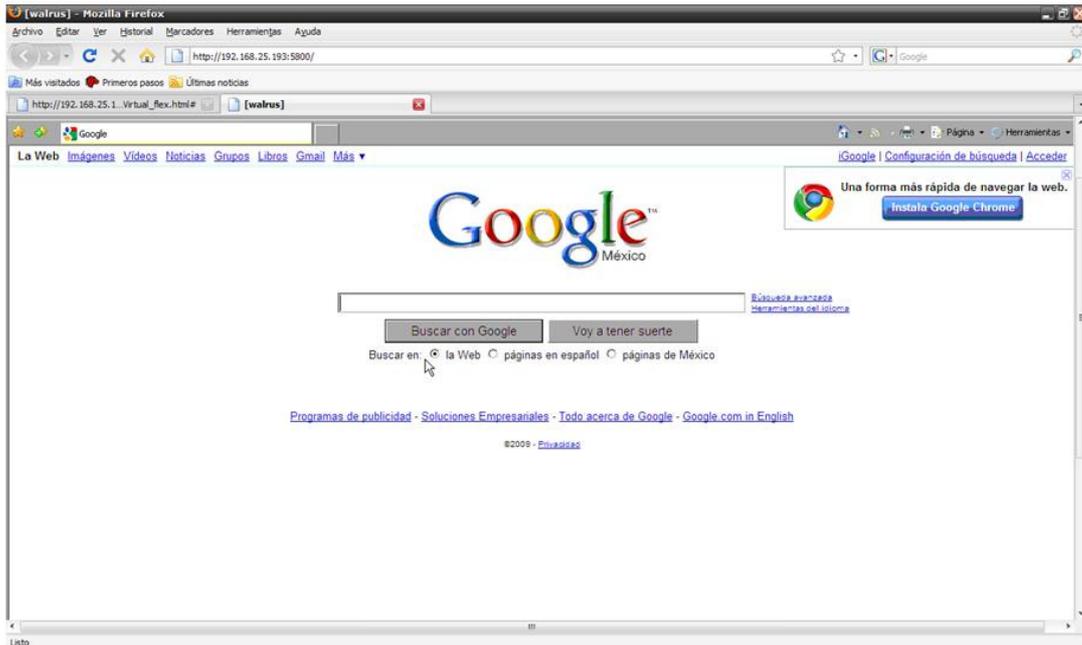


a) Componente: pizarrón

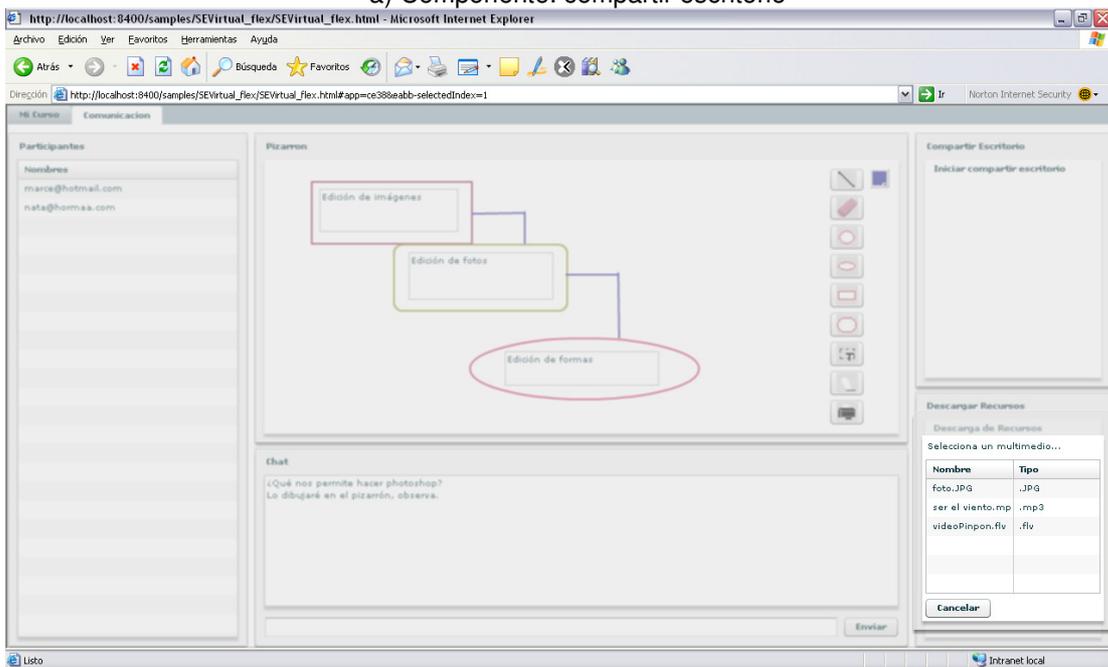
b) Componente: chat

Figura A.33. Componentes: pizarrón y chat

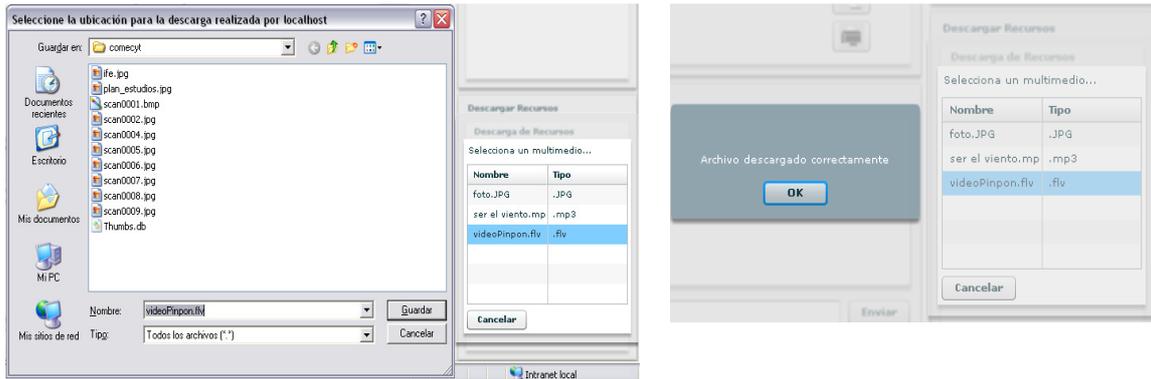
La **Figura A.34 (a)** despliega el resultado de compartir el escritorio de un participante a los demás, en este caso el participante está mostrando su navegador *Web*. La **Figura A.34 (b)** presenta el resultado de dar clic en la opción: "Descargar Recursos", la cual despliega un componente con todos los recursos que el profesor ha subido al servidor, el participante puede descargar cualquiera de los recursos. La **Figura A.34 (c)** y **(d)** muestran el resultado de descargar el recurso "videoPinpon.flv" al dar doble clic sobre el recurso y seleccionar el destino.



a) Componente: compartir escritorio



b) Componente: descargar recursos



c) Seleccionar destino

d) Descargar archivo

Figura A.34. Componentes: compartir escritorio y descargar recursos

6.5 Evaluar curso

El alumno puede realizar la evaluación del curso, para ello es necesario dar clic en el botón **“Evaluación”** de la pestaña **“Mi Curso”**. La **Figura A.35** muestra el resultado de cargar los meta datos de la evaluación en el componente. Al inicio se carga la primer pregunta y al dar clic en el botón siguiente **“>”** de la barra de navegación se carga el meta datos de la siguiente pregunta. La **Figura A.36** muestra el resultado de la operación.

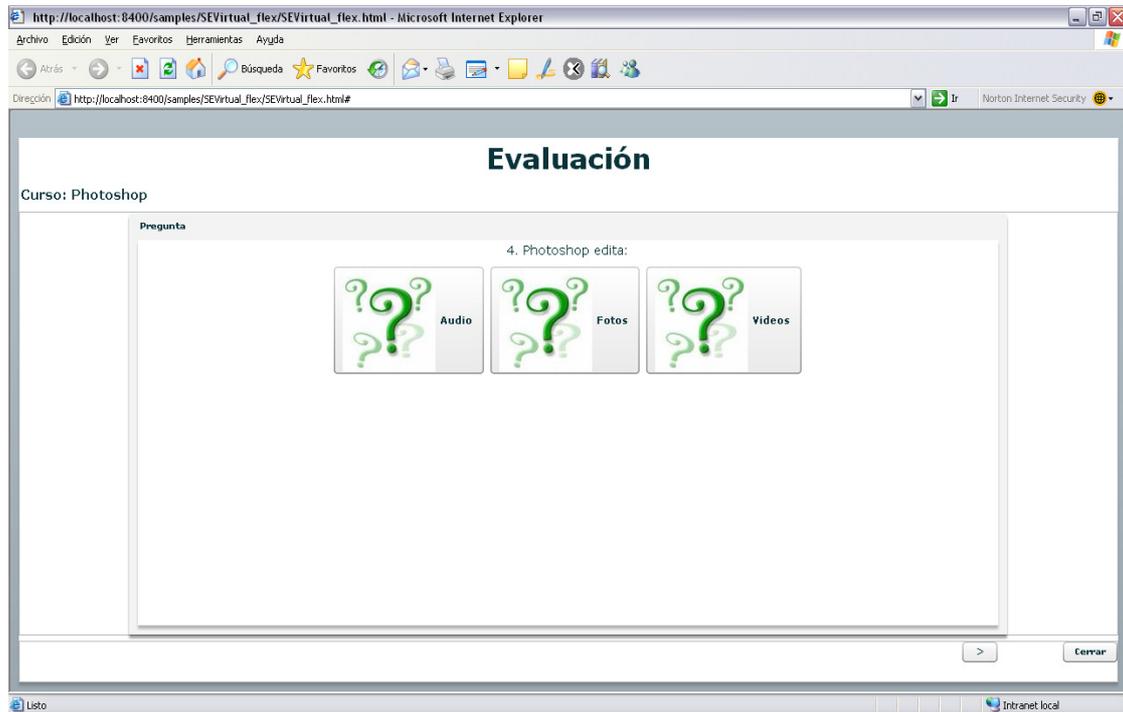


Figura A.35. Componente: evaluación

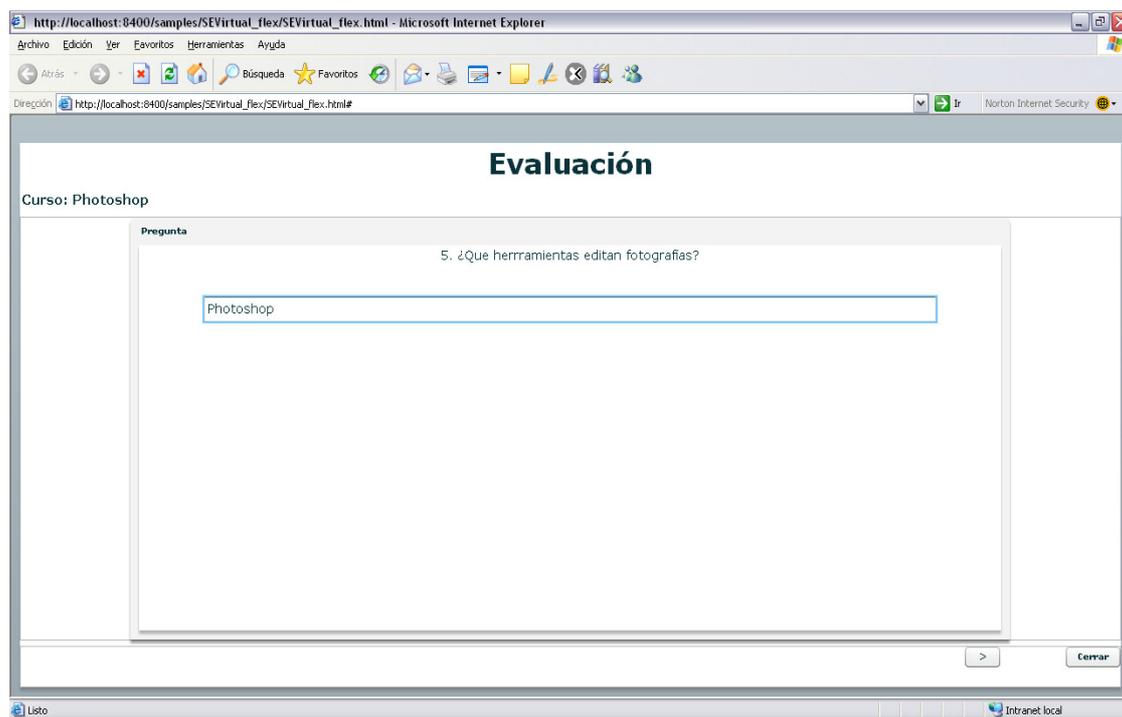


Figura A.36. Siguiete pregunta

Durante la evaluación se van enviando métricas al MAS para que al final de la evaluación el profesor virtual le dé una retroalimentación dinámica, recomendaciones y reconfiguración del curso al alumno. La **Figura A.37** despliega el resultado de la evaluación del alumno, mostrándole la calificación obtenida y una retroalimentación dinámica por parte del profesor virtual, que incluye: el nombre de la pregunta errónea y el tema al que pertenece, además al dar doble clic sobre la pregunta ésta es mostrada. La **Figura A.38** muestra la pregunta: "Photoshop edita:" que se carga al dar doble clic sobre ella. Además el profesor virtual proporciona al alumno recomendaciones relacionadas con el tema que se le dificultó, como se muestra en la **Figura A.37** donde el profesor virtual le recomienda libros, páginas *Web* y ejercicios relacionados con el tema denominado "Introducción".

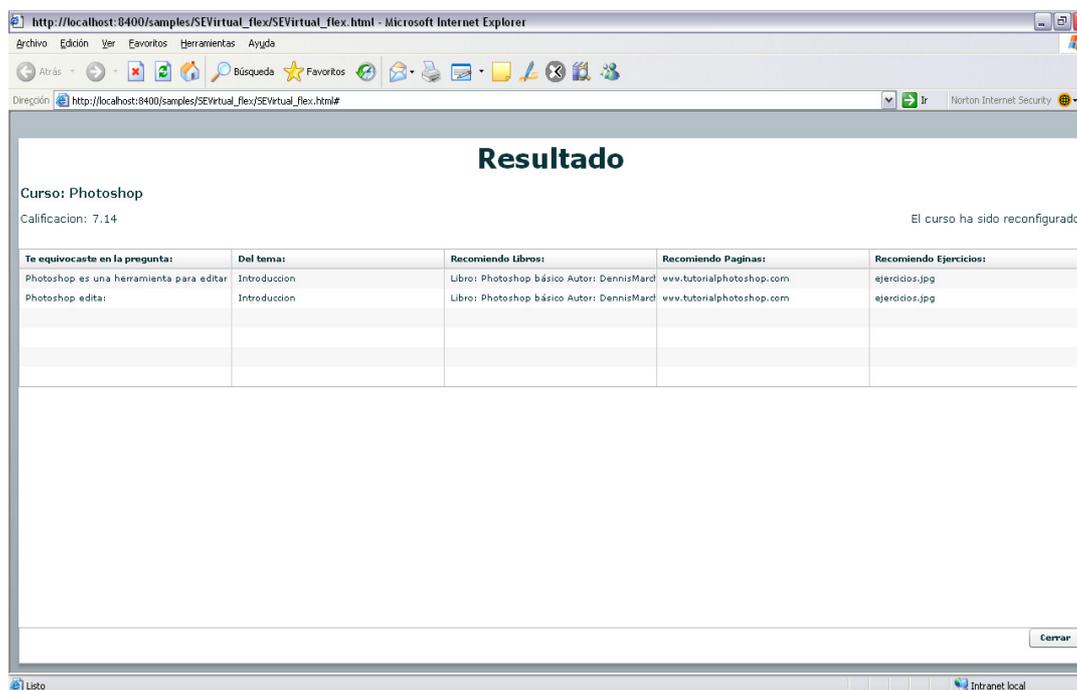


Figura A.37. Resultado de la evaluación

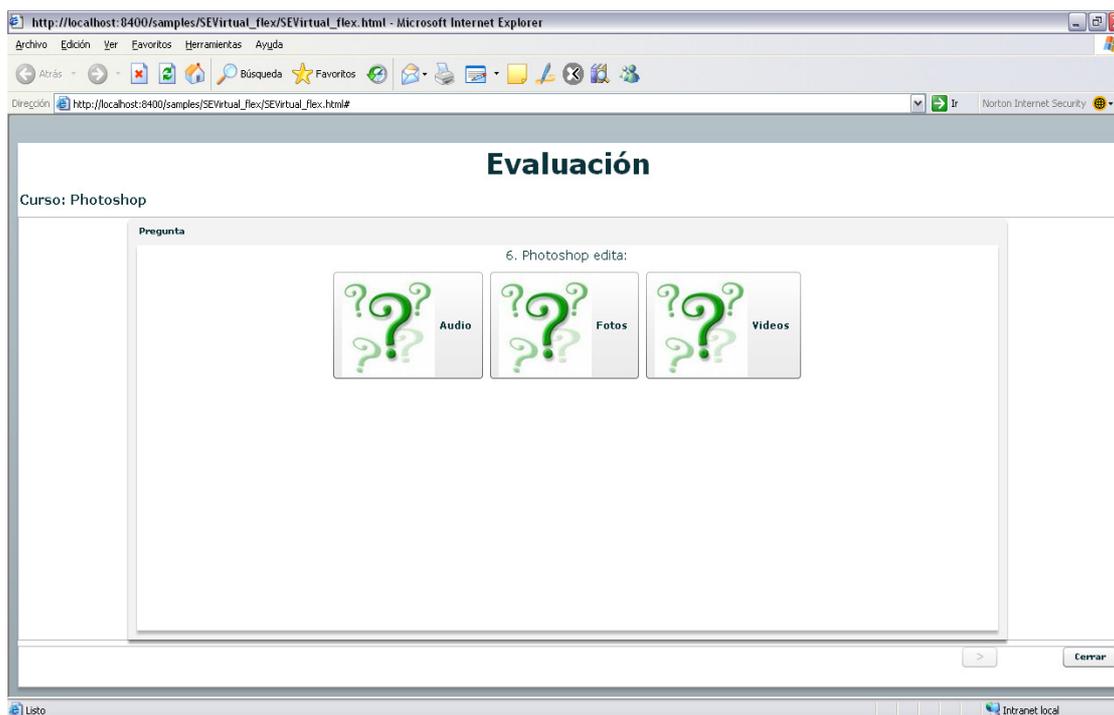


Figura A.38. Pregunta: "Photoshop edita:"

Una vez que el alumno ha obtenido el resultado de la evaluación, puede acceder al curso que el profesor virtual le ha reconfigurado, para ello es necesario ir al menú del alumno y dar clic en la opción "Cursos Reconfigurados", la **Figura A.39** muestra el resultado de esta operación, en la cual se muestra la lista de los cursos que le han sido reconfigurados.

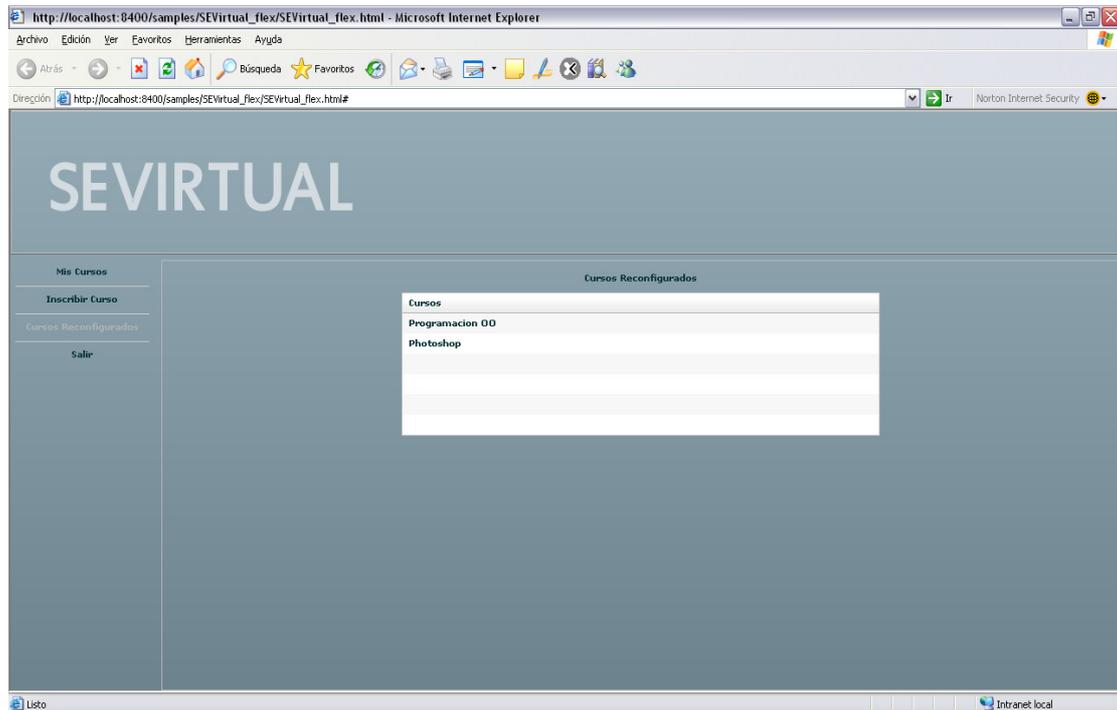


Figura A.39. Cursos reconfigurados

Para visualizar alguno de estos cursos, es necesario dar doble clic sobre el curso, la **Figura A.40** muestra el resultado de dar doble clic sobre el curso "Photoshop"; como se observa ésta solo contiene al tema "Introducción" ya que al alumno de acuerdo al resultado de su evaluación, mostrada en la evaluación solo se le dificultó este tema. El curso reconfigurado se carga de los meta datos creados por el profesor virtual.

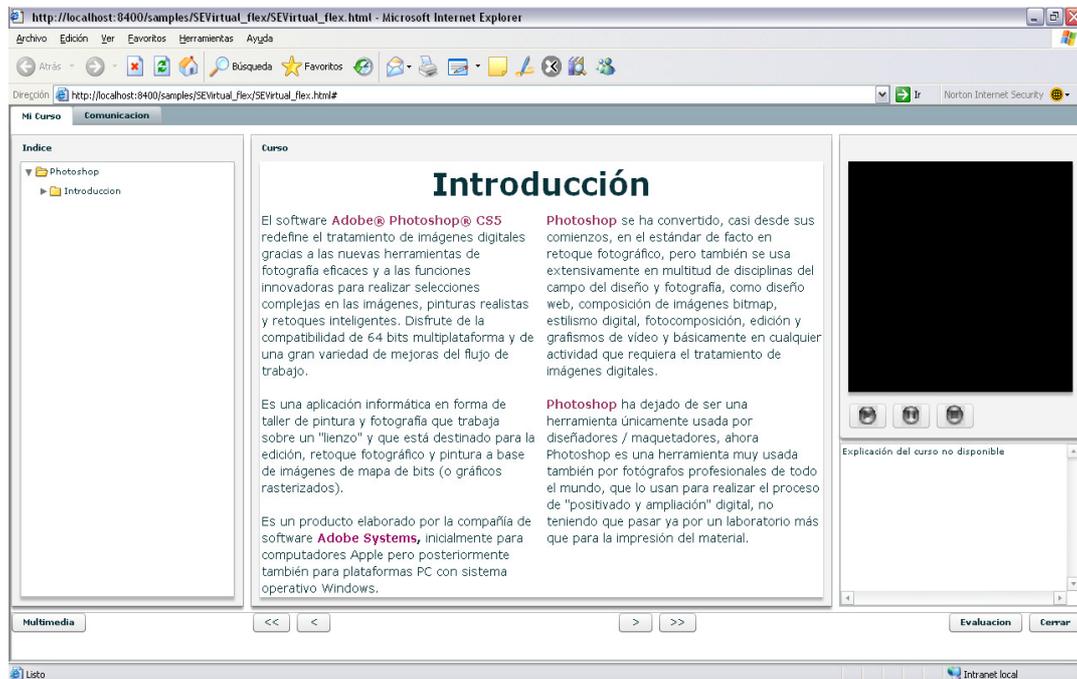


Figura A.40. Curso reconfigurado: "Photoshop"